

**TC
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**YAPAY SİNİR AĞLARININ YENİ METASEZGİSEL
ALGORİTMALAR İLE EĞİTİMİ VE VERİ
MADENCİLİĞİNDE SINIFLANDIRMA ALANINDA
KULLANIMI**

**Hazırlayan
Burak GÜLMEZ**

**Danışman
Doç. Dr. Sinem KULLUK**

Yüksek Lisans Tezi

**Ağustos 2016
KAYSERİ**

**TC
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**YAPAY SİNİR AĞLARININ YENİ METASEZGİSEL
ALGORİTMALAR İLE EĞİTİMİ VE VERİ
MADENCİLİĞİNDE SINIFLANDIRMA ALANINDA
KULLANIMI**

**Hazırlayan
Burak GÜLMEZ**

**Danışman
Doç. Dr. Sinem KULLUK**

**Bu çalışma Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi
tarafından FYL-2016-6771kodlu proje ile desteklenmiştir.**

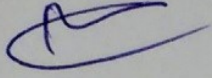
**Ağustos 2016
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Adı Soyadı: Burak GÜLMEZ

İmza:



YÖNERGEYE UYGUNLUK

“Yapay Sinir Ağlarının Yeni Metasezgisel Algoritmalar ile Eğitimi ve Veri Madenciliğinde Sınıflandırma Alanında Kullanımı” adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi 'ne uygun olarak hazırlanmıştır.

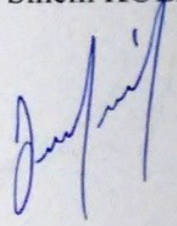
Tezi Hazırlayan

Burak GÜLMEZ



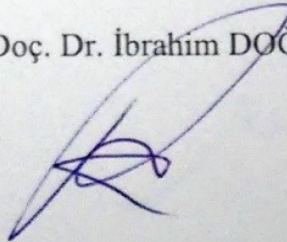
Danışman

Doç. Dr. Sinem KULLUK



Endüstri Mühendisliği ABD Başkanı

Doç. Dr. İbrahim DOĞAN



Doç. Dr. Sinem KULLUK danışmanlığında Burak GÜLMEZ tarafından hazırlanan “Yapay Sinir Ağlarının Yeni Metasezgisel Algoritmalar ile Eğitimi ve Veri Madenciliğinde Sınıflandırma Alanında Kullanımı” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

08 / 08 / 2016

JÜRİ:

Danışman: Doç. Dr. Sinem KULLUK

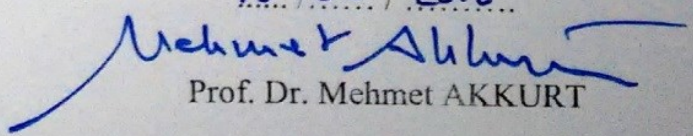
Üye : Doç. Dr. Lale ÖZBAKIR

Üye : Yrd. Doç. Dr. Hamdi Giray REŞAT

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 16/08/2016 tarih ve 2016/36-13 sayılı kararı ile onaylanmıştır.

16 / 08 / 2016


Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım boyunca benden yardımını esirgemeyen, görüşleri ve katkılarıyla beni yönlendiren, tecrübelerinden faydalandığım başta hocam Doç. Dr. Sinem KULLUK olmak üzere üzerimde emeği bulunan herkese teşekkür ederim.

FYL-2016-6771 kodlu proje ile çalışmalarımı destekleyen Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne teşekkür ederim.

Burak GÜLMEZ

Kayseri, Ağustos 2016

YAPAY SİNİR AĞLARININ YENİ METASEZGİSEL ALGORİTMALAR İLE EĞİTİMİ VE VERİ MADENCİLİĞİNDE SINIFLANDIRMA ALANINDA KULLANIMI

Burak GÜLMEZ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Ağustos 2016

Danışman: Doç. Dr. Sinem KULLUK

ÖZET

Veri madenciliği günümüzde önemini gitgide artıran bir alandır ve veri madenciliği sayesinde birçok alanda kolaylık ve gelişmeler sağlanmaktadır. Bankacılıktan eğitime, sağlıktan askeriye kadar veri madenciliğinin çok yaygın bir kullanım alanı vardır. Sınıflandırma veri madenciliğinin önemli bir parçasıdır. Sınıflandırmada eldeki veriler, belirli kurallara göre belirli sınıflara atanırlar. Sınıflandırma alanında birçok yöntem ve algoritma geliştirilmiştir. Bu yöntemler arasında yapay sinir ağları elde ettikleri yüksek doğruluk değerleri ile en çok kullanılan yöntemlerden birisidir. Yapay sinir ağları insan beyninden esinlenerek insanda bulunan yeni bilgiler öğrenme, yeni çıkarımlar yapma gibi yetenekleri otomatik olarak gerçekleştiren bir yöntemdir ve doğrusal olmayan karmaşık problemleri çözmekte oldukça başarılıdır. Bu yüzden eğitimleri yavaş ve zordur. Yapay sinir ağlarının eğitiminde kullanılan geleneksel algoritmalar matematiksel hesaplamalara dayanmaktadır, yavaştır ve genellikle yerel optimum noktaya takılıp kalmaktadırlar. Bu dezavantajların üstesinden gelen metasezgisel algoritmalar literatürde yaygın olarak kullanılmaktadır. Genetik algoritmalar, yapay arı kolonisi algoritması gibi birçok metasezgisel algoritma yapay sinir ağlarının eğitiminde kullanılmış ve oldukça etkin sonuçlar elde edilmiştir. Bu çalışmada son yıllarda geliştirilen ve sürekli problemler üzerinde oldukça iyi sonuçlar veren iyon hareketi algoritması, sosyal örümcek algoritması ve stokastik fraktal arama algoritmalarının yapay sinir ağlarının eğitimindeki başarı düzeyleri değerlendirilmiştir. Algoritmaların performansı literatür verileri ve bir gerçek hayat verisi üzerinde değerlendirilmiş ve hem geleneksel hem de metasezgisel algoritmalarla karşılaştırılmıştır.

Anahtar kelimeler: Veri madenciliği, sınıflandırma, metasezgisel algoritmalar, yapay sinir ağları

**TRAINING ARTIFICIAL NEURAL NETWORKS WITH NEW
METAHEURISTIC ALGORITHMS AND THEIR APPLICATION TO
CLASSIFICATION AREA OF DATA MINING**

Burak GÜLMEZ

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, August 2016

Supervisor: Assoc. Prof. Dr. Sinem KULLUK

ABSTRACT

Data mining is an area that gradually increases its importance nowadays and ease and development in many areas is provided through data mining. Data mining has common application area from banking to education, health to military. Classification is an important part of data mining. In classification, the data are assigned to certain classes according to certain rules. Many methods and algorithms are developed in classification. Artificial neural networks are one of the most widely used methods with high accuracy values among these methods. Artificial neural networks inspired by the human brain is a method that automatically perform abilities in humans such as learning new information, making new inferences and is very successful in solving complex nonlinear problems. Therefore, their training is slow and difficult. Traditional algorithms used in training artificial neural networks are based on mathematical computation, slow and often trapped into local optima. Metaheuristic algorithms are widely used in literature that overcomes these disadvantages. Many metaheuristic algorithms like genetic algorithms, artificial bee colony algorithm used in training artificial neural networks and very effective results were obtained. In this study, success level in training neural networks of ions motion algorithm, social spider algorithm and stochastic fractal search algorithm which were developed in recent years and given very effective results on continuous problems are evaluated. The performance of the algorithms are evaluated on benchmark problems and a real life data and compared with both traditional and metaheuristic algorithms.

Keywords: Data mining, classification, metaheuristic algorithms, artificial neural networks

İÇİNDEKİLER

BİLİMSEL ETİĞE UYGUNLUK	i
YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY SAYFASI	iii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT	vi
İÇİNDEKİLER	vii
KISALTMA VE SİMGELER.....	x
TABLolar	xii
ŞEKİLLER.....	xv
1. BÖLÜM.....	1
GİRİŞ	1
1.1 Tezin Konusu ve Kapsamı	1
1.2 Tezin Amacı ve Yöntemi	2
1.3 Tezin Bölümleri.....	3
2. BÖLÜM.....	4
VERİ MADENCİLİĞİ.....	4
2.1 Veri Madenciliği Tanımı	4
2.2 Veri Tabanından Bilgi Keşfi (VTBK).....	4
2.3 Veri Madenciliğinde Karşılaşılan Problemler	5
2.4 Veri Madenciliği Fonksiyonları	6
2.5 Veri Madenciliği Modelleri.....	8
2.5.1 Birliktelik Analizi.....	8
2.5.2 Kümeleme	8
2.5.3 Sınıflandırma ve Regresyon Modelleri	9
3. BÖLÜM.....	10
YAPAY SİNİR AĞLARI	10
3.1 Yapay Sinir Ağlarının Tanımı	10
3.2 Yapay Sinir Ağlarının Tarihçesi.....	10
3.3 Yapay Sinir Ağlarının Kullanım Alanları	12
3.4 Yapay Sinir Ağlarının Avantajları	12

3.5	Yapay Sinir Ağlarının Dezavantajları	14
3.6	Biyolojik Sinir Hücresi	15
3.7	Yapay Sinir Hücresi	16
3.8	Yapay Sinir Ağlarının Yapısı	21
3.9	Yapay Sinir Ağlarının Sınıflandırılması.....	22
3.9.1	Katman Sayısına Göre Sınıflandırma	22
3.9.2	Yapılarına Göre Sınıflandırma	22
3.9.3	Öğrenme Yöntemine Göre Sınıflandırma	23
3.10	Öğrenme Kuralları.....	23
3.11	Yapay Sinir Ağı Tasarımı.....	25
3.12	Geri Yayılım Algoritması.....	26
4.	BÖLÜM.....	29
	YAPAY SİNİR AĞLARI EĞİTİMİ LİTERATÜR TARAMASI.....	29
5.	BÖLÜM.....	34
	METASEZGİSEL ALGORİTMALAR.....	34
5.1	İyon Hareketi Algoritması.....	35
5.1.1	Sıvı Faz (Farklılaştırma)	36
5.1.2	Katı Faz (Kuvvetlendirme)	38
5.2	Sosyal Örümcek Algoritması	40
5.3	Stokastik Fraktal Arama Algoritması.....	44
5.3.1	Difüzyon İşlemi.....	44
5.3.2	Güncelleme İşlemi	44
6.	BÖLÜM.....	47
	YENİ NESİL METASEZGİSELLERLE YSA EĞİTİMİ	47
6.1	Veri Kümeleri.....	47
6.2	Çalışmada Kullanılan Yeni Metasezgiseller ve Karşılaştırmada Kullanılan Algoritmalar	48
6.3	Dizi Gösterimi	49
6.4	YSA Tasarımı ve Veri Kümelerinin Eğitim-Test Kümelerine Ayırılması.....	50
6.5	Etkinlik Ölçütleri	51
6.6	Deney Tasarımı	51
6.6.1	İyon Hareketi Algoritması Deney Tasarımı	52
6.6.2	Sosyal Örümcek Algoritması Deney Tasarımı	63

6.6.3	Stokastik Fraktal Arama Algoritması Deney Tasarımı.....	90
6.7	Yeni Metasezgisel Algoritmaların Parametre Değerleri	100
6.8	Yeni Metasezgiseller ile Elde Edilen Sonuçlar	101
6.9	Gerçek Hayat Verisi ile Yeni Nesil Metasezgisel Algoritmaların Testi	107
7.	BÖLÜM	112
	SONUÇLAR	112
	KAYNAKÇA	114
	EKLER	117
	ÖZGEÇMİŞ	149

KISALTMA VE SİMGELER

ABC	Yapay arı kolonisi algoritması (Artificial Bee Colony Algorithm)
BP	Geri yayılım algoritması (Backpropagation Algorithm)
DE	Diferansiyel gelişim algoritması (Differential Evolution Algorithm)
GA	Genetik algoritma (Geneetic Algorithm)
IM	İyon hareketi algoritması (Ion Motion Algorithm)
LM	Levenberg–Marquardt algoritması
PSO	Parçacık sürü optimizasyonu algoritması (Particle Swarm Optimization Algorithm)
SFS	Stokastik fraktal arama algoritması (Stochastic Fractal Search Algorithm)
SS	Sosyal örümcek algoritması (Social Spider Algorithm)
VM	Veri madenciliği
VTBK	Veri tabanlarından bilgi keşfi
YSA	Yapay sinir ağları
HA	Harmoni arama algoritması (Harmony Search Algorithm)
HS	Av arama algoritması (Hunting Search Algorithm)
IR	İzotonik regresyon (Isotonic Regression)
GP	Gauss süreçleri (Gaussian Processes)
PR	Tempo regresyon (Pace Regression)
LR	Doğrusal regresyon (Linear Regression)

RBFN	Radyal tabanlı fonksiyon ağı (Radial Basis Fuction Network)
LeastMedSq	En düşük medyan karesi doğrusal regresyon (Least Median Squared Linear Regression)
SMOrg	Destek vektör makinesi (Support Vector Machine)
MLP	Çok katmanlı algılayıcı (Multi Layer Perceptron)

TABLOLAR

Tablo 3.1 Biyolojik sinir sistemi ve yapay sinir sistemindeki karşılıkları.....	16
Tablo 5.1 Kristal faz sözde kodu.....	38
Tablo 5.2 Sosyal örümcek algoritması sözde kodu.....	43
Tablo 5.3 Stokastik fraktal arama algoritması sözde kodu	46
Tablo 6.1 Veri kümelerinin temel özellikleri.....	47
Tablo 6.2 YSA eğitimi için tutulan ağırlık matrisleri	49
Tablo 6.3 İyon hareketi algoritması deney tasarımı parametreleri	52
Tablo 6.4 IM algoritması Gene veri kümesi deney tasarımı sonuçları	53
Tablo 6.5 IM algoritması Cancer veri kümesi deney tasarımı sonuçları	56
Tablo 6.6 IM algoritması Card veri kümesi deney tasarımı sonuçları.....	56
Tablo 6.7 IM algoritması Diabetes veri kümesi deney tasarımı sonuçları.....	58
Tablo 6.8 IM algoritması Glass veri kümesi deney tasarımı sonuçları.....	59
Tablo 6.9 IM algoritması Soybean veri kümesi deney tasarımı sonuçları.....	61
Tablo 6.10 IM algoritması Thyroid veri kümesi deney tasarımı sonuçları.....	62
Tablo 6.11 Sosyal örümcek algoritması deney tasarımı parametreleri	63
Tablo 6.12 SS algoritması Cancer veri kümesi deney tasarımı sonuçları.....	64
Tablo 6.13 SS algoritması Card veri kümesi deney tasarımı sonuçları	69
Tablo 6.14 SS algoritması Diabetes veri kümesi deney tasarımı sonuçları	73
Tablo 6.15 SS algoritması Gene veri kümesi deney tasarımı sonuçları.....	77

Tablo 6.16 SS algoritması Glass veri kümesi deney tasarımı sonuçları	81
Tablo 6.17 SS algoritması Soybean veri kümesi deney tasarımı sonuçları	85
Tablo 6.18 SS algoritması Thyroid veri kümesi deney tasarımı sonuçları	89
Tablo 6.19 SFS algoritması deney tasarımı parametreleri	90
Tablo 6.20 SFS algoritması Gene veri kümesi deney tasarımı sonuçları	91
Tablo 6.21 SFS algoritması Cancer veri kümesi deney tasarımı sonuçları	95
Tablo 6.22 SFS algoritması Card veri kümesi deney tasarımı sonuçları	95
Tablo 6.23 SFS algoritması Diabetes veri kümesi deney tasarımı sonuçları	97
Tablo 6.24 SFS algoritması Glass veri kümesi deney tasarımı sonuçları	98
Tablo 6.25 SFS algoritması Soybean veri kümesi deney tasarımı sonuçları	99
Tablo 6.26 SFS algoritması Thyroid veri kümesi deney tasarımı sonuçları	100
Tablo 6.27 IM algoritması parametre değerleri	101
Tablo 6.28 SS algoritması parametre değerleri	101
Tablo 6.29 SFS algoritması parametre değerleri	101
Tablo 6.30 IM, SS ve SFS algoritmalarının elde ettiği sonuçlar	102
Tablo 6.31 Eşleştirmeli-t testi p değerleri	103
Tablo 6.32 Karşılaştırma Sonuçları	104
Tablo 6.33 Ortalama CEP değerleri için 1-örnek t testi p değerleri	105
Tablo 6.34 Üretim veri kümesi değişken tanımları	108
Tablo 6.35 Algoritmaların üretim verisi üzerindeki test sonuçları	109

Tablo 6.36 Eşleştirilmiş-t testi p değerleri	110
Tablo 6.37 Karşılaştırma Sonuçları.....	110
Tablo 6.38 RMS değerleri için Eşleştirmeli-t testi p değerleri	111

ŞEKİLLER

Şekil 2.1 VTBK süreci	5
Şekil 3.1 Biyolojik sinir hücresi	15
Şekil 3.2 Yapay sinir hücresi	16
Şekil 3.3 Sigmoid fonksiyonu	18
Şekil 3.4 Doğrusal fonksiyon	19
Şekil 3.5 Basamak fonksiyonu	19
Şekil 3.6 Hiperbolik tanjant fonksiyonu	20
Şekil 3.7 Sinüs fonksiyonu	20
Şekil 3.8 Yapay sinir ağlarının yapısı	21
Şekil 3.9 a) İleri besleme b) Geri besleme	26
Şekil 5.1 Anyon ve katyonların hareketi	36
Şekil 5.2 Anyon ve katyonların en iyi anyon ve katyona doğru yönelimleri	37
Şekil 5.3 Yerel optimuma takılan iyonların dağıtılması	39
Şekil 5.4 İyon hareketi algoritması akış şeması	40
Şekil 6.1 YSA eğitimi için tutulan ağırlık değerleri matrisleri	50
Şekil 6.2 IM algoritması Gene veri kümesi normal olasılık eğrisi	53
Şekil 6.3 IM algoritması Gene veri kümesi pareto grafiği	54
Şekil 6.4 IM algoritması Gene veri kümesi tekli parametre etkileri	54
Şekil 6.5 IM algoritması Gene veri kümesi ikili parametre etkileri	55

Şekil 6.6 IM algoritması Card veri kümesi tekli parametre etkileri.....	57
Şekil 6.7 IM algoritması Diabetes veri kümesi tekli parametre etkileri	58
Şekil 6.8 IM algoritması Glass veri kümesi tekli parametre etkileri.....	60
Şekil 6.9 IM algoritması Soybean veri kümesi tekli parametre etkileri.....	61
Şekil 6.10 SS algoritması Cancer veri kümesi normal olasılık eğrisi	65
Şekil 6.11 SS algoritması Cancer veri kümesi pareto grafiği	65
Şekil 6.12 SS algoritması Cancer veri kümesi tekli parametre etkileri.....	66
Şekil 6.13 SS algoritması Cancer veri kümesi ikili parametre etkileri	67
Şekil 6.14 SS algoritması Cancer veri kümesi küp grafiği	68
Şekil 6.15 SS algoritması Card veri kümesi tekli parametre etkileri	70
Şekil 6.16 SS algoritması Card veri kümesi ikili parametre etkileri.....	71
Şekil 6.17 SS algoritması Card veri kümesi küp grafiği	72
Şekil 6.18 SS algoritması Diabetes veri kümesi tekli parametre etkileri.....	74
Şekil 6.19 SS algoritması Diabetes veri kümesi ikili parametre etkileşimleri.....	75
Şekil 6.20 SS algoritması Diabetes veri kümesi küp grafiği.....	76
Şekil 6.21 SS algoritması Gene veri kümesi tekli parametre etkileri	78
Şekil 6.22 SS algoritması Gene veri kümesi ikili parametre etkileşimleri	79
Şekil 6.23 SS algoritması Gene veri kümesi küp grafiği	79
Şekil 6.24 SS algoritması Glass veri kümesi tekli parametre etkileri	82
Şekil 6.25 SS algoritması Glass veri kümesi ikili parametre etkileşimleri	83

Şekil 6.26 SS algoritması Glass veri kümesi küp grafiği	83
Şekil 6.27 SS algoritması Soybean veri kümesi tekli parametre etkileri	86
Şekil 6.28 SS algoritması Soybean veri kümesi ikili parametre etkileşimleri	87
Şekil 6.29 SS algoritması Soybean veri kümesi küp grafiği	88
Şekil 6.30 SFS algoritması Gene veri kümesi normal olasılık eğrisi	91
Şekil 6.31 SFS algoritması Gene veri kümesi pareto grafiği	92
Şekil 6.32 SFS algoritması Gene veri kümesi tekli parametre etkileri	92
Şekil 6.33 SFS algoritması Gene veri kümesi ikili parametre etkileşimleri	93
Şekil 6.34 SFS algoritması Gene veri kümesi küp grafiği	94
Şekil 6.35 SFS algoritması Card veri kümesi tekli parametre etkileri	96
Şekil 6.36 SFS algoritması Glass veri kümesi tekli parametre etkileri	98

1. BÖLÜM

GİRİŞ

1.1 Tezin Konusu ve Kapsamı

Günümüzde bilgisayar ve bilgisayarlı sistemler oldukça yaygın olarak kullanılmaktadır. Bu sistemler sadece verilen basit komutları yerine getirmekle kalmaz, ayrıca yapay zekâ ile gelişmiş ve karmaşık işlemleri gerçekleştirebilirler. Yapay zekânın yaklaşık 60 yıl öncesine dayanan bir geçmişi vardır. Zaman içinde oldukça popülerleşmiş ve büyük bir ilerleme kaydetmiştir. Günümüzde akıllı sistemler olarak adlandırılan sistemler gelişmiş bir yapay zekâyâ sahiptir. Bu sayede sadece basit hesaplamalar dışına çıkarak çeşitli algoritmalar ile büyük ve zor problemler çözülebilmektedir. Ayrıca insanın yapısına benzer şekilde tecrübe ve deneyimlerden yararlanma, öğrenme, problem çözme gibi özellikleri de başarabilmektedir [1].

Yapay sinir ağları (YSA) insan beyninden esinlenerek insanda bulunan yeni bilgiler öğrenme, yeni çıkarımlar yapma gibi yetenekleri otomatik olarak gerçekleştiren bir yöntemdir. YSA genelleme, sınıflandırma, ilişkilendirme, kural çıkarma, optimizasyon gibi amaçlarla kullanılabilir. YSA'nın çeşitli özellikleri vardır. Geleneksel programlama yöntemlerine benzememektedir, farklı bir bilgi işleme metodu vardır. Bilgiyi herhangi bir veri tabanında değil kendi içinde bağlantılarda saklar. Öğrenmek için bazı örnekleri kullanır, örnekler ile ağdaki bağlantı değerleri değişir ve ağ öğrenmeye başlar. Bu sayede gösterilmemiş örnekler hakkında tahminler yapar. YSA, eksik bilgi ile çalışabilmektedir, bu özellik yapay sinir ağlarını diğer sistemlere göre oldukça avantajlı bir hale getirmektedir. Hataya karşı toleranslıdır, örneklerde bulunan hataları tolere edebilir. YSA'da bilgi ağa yayılmıştır ve bağlantılardaki değerler olarak saklanmaktadır. Sadece bu değerlere bakılarak bilgiler açıklanamaz, yorumlanamaz.

Yapay sinir ağıları hem sürekli hem de kesikli verilerle çalışabilmektedir, sözel veriler ise sınıflandırılarak veya kesiklendirilerek kullanılabilir [2].

Yapay sinir ağıları zor ve karmaşık farklı türlerdeki problemlere uygulanabilmektedir. Özellikle çok değişkenli, değişkenler arasında etkileşimli, tam bir çözümün bulunmaması, farklı çözümlerin mümkün olması gibi problemlerde kullanılabilir. Tıp, görüntü tanıma sistemleri, parmak izi tespit sistemleri, kalite kontrolü, üretim, ekonomi, haberleşme gibi çok Geniş bir uygulama kapsamına sahiptir ve bu alanlarda kullanılabilir. Veri madenciliğinde (VM) sınıflandırma alanında çok sayıda araştırma yapılmaktadır. Bu araştırmalardan birçoğu metasezgisel algoritmalarla yapay sinir ağlarının eğitimi üzerinedir. Bu sayede yapay sinir ağıları kullanılarak sınıflandırmalarda etkili sonuçlar elde edilmektedir [2].

1.2 Tezin Amacı ve Yöntemi

Yapay sinir ağıları sınıflandırma problemlerinde kullanılan yöntemlerden birisidir. Bir yapay sinir ağını kullanmak ve etkili sonuçlar elde etmek için eğitmek gerekmektedir. Bu eğitim matematiksel hesaplara dayanan geleneksel yöntemlerle veya metasezgisel algoritmalarla yapılabilir. Metasezgisel algoritmalar diğer yöntemlerden hız, zaman ve iyi sonuç elde etmek bakımından daha avantajlıdır. Bu çalışmada yapay sinir ağlarının eğitiminde yeni metasezgisel algoritmaların kullanılmasıyla geleneksel yöntemlerden daha iyi sonuçlar elde edilmesi amaçlanmaktadır. Literatür araştırmaları sonucunda yapay sinir ağlarının eğitimi için henüz kullanılmamış üç yeni nesil metasezgisel algoritmanın yapay sinir ağlarının eğitiminde kullanılmasıyla literatüre de katkı sağlanması hedeflenmiştir.

Yeni nesil metasezgiseller ile yapay sinir ağlarının eğitimi için algoritmalar bilgisayar ortamında Java programlama diliyle kodlanmıştır. Kodlanan bu algoritmaların test edilebilmesi için literatürdeki veriler ve bir gerçek hayat verisi kullanılmıştır. Literatürde sınıflandırma problemleriyle ilgili çok sayıda veri mevcuttur. Bu veriler üzerinde test edilen algoritmalar hem geçmişteki çalışmalarla hem de kendi aralarında karşılaştırılmıştır. Gerçek hayat probleminde elde edilen sonuçlar ise geleneksel yöntemler ve ele alınan algoritmalar ile kendi aralarında karşılaştırılmıştır.

1.3 Tezin Bölümleri

Tez çalışmasının ilk bölümünde tez ile alakalı genel bir giriş yapılmıştır. Tezin konusu, amacı ve yönteminden bahsedilmiştir.

İkinci bölümde veri madenciliği ve sınıflandırmadan genel hatları ile bahsedilmiştir.

Üçüncü bölümde yapay sinir ağları hakkında bilgiler verilmiştir. Yapay sinir ağlarının tarihçesi, kullanım alanları, avantajları, yapısı gibi konulardan bahsedilmiştir.

Dördüncü bölümde yapay sinir ağları eğitimiyle alakalı literatür araştırmasına yer verilmiş ve bu konudaki geçmiş çalışmalardan bahsedilmiştir.

Beşinci bölümde bu çalışmada kullanılan yeni nesil metasezgisel algoritmalar ile alakalı bilgiler verilmiştir. Çalışmada yapay sinir ağı eğitiminde kullanılan algoritmalar detaylı bir şekilde açıklanmıştır.

Altıncı bölümde yeni metasezgisel algoritmalar ile yapay sinir ağlarının eğitimi için deneysel tasarım çalışması sunulmuş, belirlenen parametre kümeleri ile algoritmalar eğitilmiş ve elde edilen sonuçlar karşılaştırılarak sunulmuştur. Ayrıca, algoritmalar arasındaki performans farklılıklarının istatistiksel olarak anlamlı olup olmadığını belirlemek adına istatistiksel çalışmalara da bu bölümde yer verilmiştir.

Son bölüm olan sonuç bölümünde ise tez çalışması ile elde edilen sonuçlar literatür çalışmaları ışığında yorumlanmış, konu ile ilgili gelecekte yapılabilecek çalışmalarla ilgili önerilerde bulunulmuştur.

2. BÖLÜM

VERİ MADENCİLİĞİ

2.1 Veri Madenciliği Tanımı

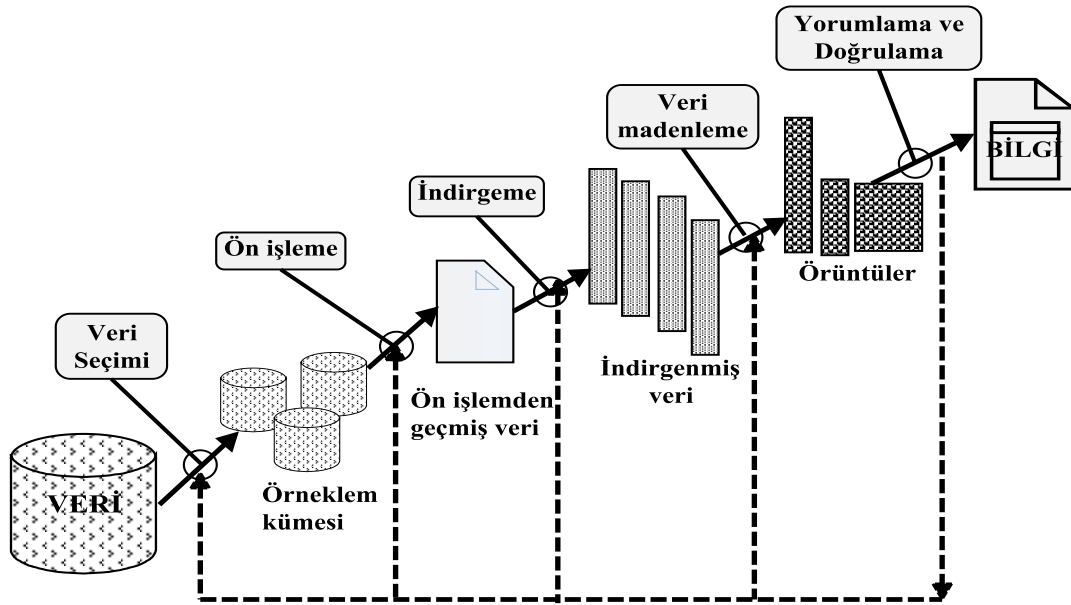
Son yıllarda bilgisayarların ve veri depolama sistemlerinin gelişmesiyle birlikte büyük miktardaki verilere kolaylıkla ulaşılabilmektedir. Veri madenciliği bu büyük miktarlardaki verilerin işlenip, analiz edilip bilgiye çevrilmesi olarak tanımlanabilir. Veri madenciliği sayesinde karar almadan önce verilerden elde edilen bilgiler bulunur ve bu bilgiler ışığında daha kolay ve etkili kararlar alınabilir.

2.2 Veri Tabanından Bilgi Keşfi (VTBK)

Genel olarak veri tabanlarından bilgi keşfi, veride mevcut olan doğru, yeni, kullanışlı ve sonuç olarak anlaşılır örüntülerin belirlenmesi olarak tanımlanır. VTBK, veri analizinden ve veride mevcut örüntülerin keşfinden daha fazla anlamlar ifade eder. VTBK, verinin nasıl saklanması ve algoritmaların büyük veri kümelerine nasıl uyarlanması gerektiği, sonuçların nasıl yorumlanacağı ve görselleştirileceği, insan-makine etkileşiminin nasıl modelleneceği sorularının cevabını aramaktadır.

VTBK, veride mevcut bilginin bulunması sürecinin tamamını kapsar. VM ise bu süreçteki merkezi adımdır ve verilerden modeller veya örnekler üretmek için veri analizleri ve keşif algoritmaları kullanmayı gerektirir.

VTBK süreci birkaç aşamadan oluşan etkileşimli ve iteratif bir süreçtir. Bu aşamalar veri seçimi, veri temizleme ve ön işleme, veri indirgeme, veri madenciliği ve değerlendirme aşamalarıdır [3]. Şekil 2.1 VTBK sürecini göstermektedir.



Şekil 2.1 VTBK süreci

Veri seçimi, veri kümesinden uygun örneklem kümesinin elde edildiği adımdır. Veri temizleme ve önileme, seçilen örneklemdeki hatalı örneklerin çıkarıldığı ve eksik nitelik değerlerinin değiştirildiği aşamadır. Bu aşamada keşfedilen bilginin kalitesi artırılır. Veri indirgeme, seçilen örneklemdeki alakasız niteliklerin atıldığı ve tekrarlı örneklerin saptandığı aşamadır. Bu sayede, veri madenciliğinin çalışma süresi kısaltılır ve daha iyi bir performansta çalışması sağlanır. Veri madenciliği, belirlenen veri madenciliği işleminin yapılmasıdır. Değerlendirme ve sonuç, sonuçların ve keşfedilen bilgilerin değerlendirilme aşamasıdır.

2.3 Veri Madenciliğinde Karşılaşılan Problemler

Veri madenciliğinde büyük boyutlu ve gerçek hayat verileriyle çalışılmaktadır. Bu yüzden VM’de birçok problemle başa çıkmak gerekmektedir. Bu problemler şu şekilde özetlenebilir [3]:

Veri tabanı boyutu: veri tabanı boyutları teknolojinin gelişmesiyle büyük bir hızla artmaktadır. Hem özellik sayısı olarak sütun bazında hem de örneklem sayısı olarak satır bazında artmaktadır. Büyük bir veri tabanıyla çalışmak, ondan anlamlı bilgiler elde etmek oldukça zordur. Bununla başa çıkmak için veri kümesinin indirgenmesi veya

kullanılan algoritmaların büyük boyutlarla başa çıkabilecek kadar iyi seçilmesi gibi yollara başvurulabilir.

Gürültülü veri: veri girişi sırasında oluşan hatalar gürültülü veriye sebep olur. Bu hatalar sonucunda birçok değer yanlış olabilir. Özellikle gerçek hayat verileri düşünüldüğünde bu oldukça kötü bir durumdur. Bundan dolayı gürültülü verilerin saptanması veya etkilerinin en aza indirilmesi önemli bir problemdir.

Boş veriler: veri tabanında bulunan boş değerlerdir. Bilinmeyen bir değeri ifade eder. Bu bilinmeyen değerleri çalışmaya dâhil etmek için çeşitli yöntemler kullanılabilir. Boş değerleri dikkate alamamak, boş değerlerin yerine en çok rastlanan değer yazılması, boş değerlerin yerine ortalama bir değer yazılması, boş değer yerine varsayılan başka bir değer konulması gibi çözümler mevcuttur.

Eksik veri: veri tabanında bulunmayan verilerdir. Bu veriler sütun bazlı olabilir. Yani kayıt tutulması gereken önemli bir özellik tutulmamış olabilir. Ayrıca satır bazlı da olabilir. Belirli bir veri kümesinin veri tabanında bulunmaması sonucu oluşan eksik verilerdir.

Artık veri: veri kümesindeki gereksiz yere tutulan özellikler artık veri olarak adlandırılır. Bu verilerin elenmesi için özellik seçimi algoritmaları kullanılabilir.

Dinamik veri: içeriği sürekli değişen veri tabanları dinamik veri içerir. İçeriğin sürekli değişmesi veri madenciliği uygulamalarında problemlere neden olabilir. Veri tabanı ve veri madenciliği uygulamaları aynı anda çalışır ve bundan dolayı da bazı problemler çıkabilir.

2.4 Veri Madenciliği Fonksiyonları

Veri madenciliği fonksiyonları verilerin nasıl kullanılacağı ve verilerden elde edilmek istenen sonuçların türüne göre değişkenlik gösterir. Genel olarak veri madenciliği fonksiyonları tanımlayıcı ve tahminleyici olmak üzere ikiye ayrılabilir.

Tanımlayıcı veri madenciliğinde veri kümesi tanımlanır ve verilerin genel özellikleri sunulur. Tahminleyici veri madenciliğinde ise modeller kümesi oluşturulur mevcut

verilerden sonuçlar çıkartılır ve yeni veri kümelerinin davranışları tahmin edilmeye çalışılır. Veri madenciliğinin başlıca fonksiyonları aşağıdaki gibidir [3].

- Sınıf tanımı
- Birliktelik analizi
- Sınıflandırma
- Tahminleme
- Kümeleme
- Zaman serileri analizi
- Aykırılık analizi
- Evrimsel analiz

*Sınıf tanımı*nda veriler ortak özelliklerine göre genelleştirilmiş sınıflara ayrılırlar. Sınıf tanımı veri kümesinin kısa bir özetidir ve diğer sınıflardan farklılığını ortaya çıkarır.

Birliktelik analizi veri kümesinde sıkça birlikte görülen değerlerin saptanmasıdır. Sepet analizi olarak da adlandırılır.

Sınıflandırma bir veri kümesindeki verileri daha önceden belirlenmiş olan sınıflara dağıtma işlemidir. Sınıflandırma algoritmaları eğitim kümesinden verileri ve ait oldukları sınıfları öğrenirler. Daha sonra test kümesi ile yeni verileri sınıflara ayırırlar.

Tahminlemede bazı eksik verilerin olası değerlerinin veya değer dağılımlarını tahmin edilir.

Kümeleme verilerin birbirlerine benzerliklerine ve farklılıklarına göre kümelere ayrılmasına denir. Küme sayısı önceden belirlenebildiği gibi belirlenmeyebilir de. Kümeleme sınıflandırmaya benzemektedir fakat sınıflandırmada sınıflar önceden belirlenmiştir, kümelemede ise belirlenen herhangi bir küme yoktur, algoritma sonucu kümeler ortaya çıkar. Kümelemede ayrıca sınıflandırmadaki gibi doğruluk hesabı da yapılamaz.

Aykırılık analizi veri kümesi içerisindeki o veri kümesine uymayan aykırı olarak nitelendirilebilecek verilerin saptanması işlemidir. Bu aykırı veriler veri kümesinden çıkarılabilir veya veri kümesine uygun hale getirilebilir.

Evrimsel analizde zamanla özellikleri değişen verilerin değişim oranları ve türlerinin saptanması amaçlanmaktadır.

2.5 Veri Madenciliği Modelleri

Veri madenciliği genel olarak üç modele ayrılabilir:

- Birliktelik analizi
- Kümeleme
- Sınıflandırma ve regresyon modelleri

Bunlar arasında sınıflandırma tahmin edici bir modeldir, birliktelik analizi ve kümeleme ise tanımlayıcı modellerdir.

2.5.1 Birliktelik Analizi

Birliktelik analizi veri madenciliğinde sıkça kullanılan yöntemlerden birisidir. Sepet analizi de denir. Alışverişlerde müşterilerin satın alma alışkanlıklarını analiz edip hangi ürünleri birlikte aldıklarını belirler. Bu sayede müşterilerin daha fazla ürün satın alması amaçlanmaktadır.

2.5.2 Kümeleme

Kümeleme veriyi daha önceden belirlenmeyen kümelere ayırma işlemine denir. Aynı kümedeki elemanlar benzer özelliklere sahip oldukları gibi farklı kümelerdeki elemanlar birbirlerine pek benzemezler. Sınıflandırmadan farklıdır. Sınıflandırmada sınıflar daha önceden belirlenmiş iken kümelemede kümeler belli değildir.

Kümeleme için pek çok algoritma vardır. Genel olarak şu şekilde sıralanabilirler [4]:

- Bölme yöntemleri
- Hiyerarşik yöntemler
- Yoğunluk tabanlı yöntemler
- Izgara tabanlı yöntemler
- Model tabanlı yöntemler

Kullanım amaçlarına ve kullanım alanlarına göre kümeleme analizinin amaçları şu şekilde tanımlanabilir [5]:

- Doğru tiplerin belirlenmesi
- Model oluşturmak
- Gruplara dayalı tahmin
- Hipotez testi
- Veri araştırma
- Hipotez oluşturma
- Veri indirgeme

2.5.3 Sınıflandırma ve Regresyon Modelleri

Sınıflandırma ve regresyon modelleri veri madenciliğinde en çok kullanılan modellerdir. Sınıflandırmada veriler belirli sınıflara ayrılırlar veya regresyon modellerinde hedef değişkenle ilgili sürekli değerler bilinmektedir. Bu modeller, insan düşünce yapısına benzemektedir. Bilinmeyen veri örnekleri incelenip bu örneklerin hangi sınıfa ait olduklarına veya alacakları değerlere karar verilmektedir. Sınıfların sayısı ve tanımlamaları daha önceden bilinmektedir ve belirlenmiştir. Sınıflandırma danışmanlı bir öğrenme metodudur. Daha çok tahminlemede kullanılır [6].

Sınıflandırma ve regresyon modellerinde kullanılan algoritmalar genellikle şunlardır [3]:

- Karar ağaçları
- Yapay sinir ağları
- Evrimsel algoritmalar
- K en yakın komşu algoritması
- Bayes sınıflandırıcılar
- Doğrusal regresyon
- Doğrusal olmayan regresyon
- Sürü zekâsı teknikleri
- Kaba küme teorisi
- Bulanık küme

3. BÖLÜM

YAPAY SİNİR AĞLARI

3.1 Yapay Sinir Ağlarının Tanımı

Yapay sinir ağları insan beynindeki sinirlerin çalışma prensibinden esinlenilerek öğrenme, hatırlama, tecrübe edinme, tahmin yürütme gibi yetenekleri kullanmaya imkân sağlayan bir yöntemdir. YSA'nın literatürde farklı tanımlamaları mevcuttur:

Yapay sinir ağları insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme, keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacıyla geliştirilen bilgisayar sistemleridir [2].

Yapay sinir ağları en kısa ve basit şekilde bir örnekler kümesi yardımıyla parametrelerin uygulanabilmesini sağlayacak bir matematiksel formül için yazılan bilgisayar programı olarak tanımlanabilir. Bu tanım yapay sinir ağlarını en basit şekilde ve teknik detaya girmeden ifade etmektedir [7].

Yapay sinir ağları beyinde var olan sinirlerin çalışma prensiplerinden esinlenerek sistemlere öğrenme, bilgi edinme, bilgiler arası bağlantı ve ilişki kurma, hatırlama gibi yetenekleri kazandırmayı amaçlayan bilgi algoritmalarıdır [8].

3.2 Yapay Sinir Ağlarının Tarihçesi

Yapay sinir ağları ile ilgili çalışmalar 1900lü yılların başından günümüze kadar sürmektedir. Bu süreçteki bazı önemli olaylar ve gelişmeler aşağıda sıralanmıştır [2], [8].

1943 yılında bir nörolog olan McCulloch ve bir matematikçi olan Pitts tarafından ilk yapay sinir ağı modeli geliştirilmiştir. İnsan beyninin hesaplama yeteneğinden esinlenerek elektrik devreleriyle basit bir sinir ağı modeli geliştirmişlerdir.

1949 yılında Donald Hebb tarafından birçok öğrenme kuralının temelini oluşturan Hebb öğrenme kuralı ortaya çıkarılmıştır. 1958 yılında Rosenblatt tarafından tek katmanlı algılayıcı (perceptron) geliştirilmiştir. Perception tek katmanlı, eğitilebilen ve tek çıkışa sahip olan bir yapay sinir ağıdır.

1959 yılında Widrow ve Holf tarafından ADALINE modeli ortaya çıkartılmıştır. Bu model yapay sinir ağlarının mühendislik alanlarında kullanılması için önemli bir aşamadır. ADALINE 70lerin sonlarına doğru ortaya çıkan ve ADALINE modelinin çok katmanlı hali olan MADALINE'nin temelini oluşturmaktadır. Ayrıca gerçek dünya problemine uygulanmış ilk yapay sinir ağıdır ve telefon hatlarında oluşan yankıları yok etmeyi amaçlayan bir sistemde kullanılmıştır.

1969 yılında Minsky ve Pappert tarafından algılayıcı modelin herhangi bir değerinin olmadığı, sadece doğrusal problemler için çözüm ürettiği, doğrusal olmayan problemleri çözemediği iddia edilmiştir. İspat olarak ise XOR probleminin çözülememesini göstermişlerdir. Bu iddia genel olarak kabul görmüştür ve XOR probleminin çözülememesi YSA ile alakalı çalışmaların durmasına sebep olmuştur. Araştırma ve geliştirme faaliyetlerini yürüten ve destekleyen kuruluşlar yapay sinir ağları alanındaki çalışmalara olan destekleri kesip diğer alanlara kaydırmışlardır. Bu yüzden yapay sinir ağları ile ilgili çalışmalar XOR problemi çözülmünceye kadar sekteye uğramıştır.

1972 yılında Kohonen tarafından Korelasyon Matris Belleği geliştirilmiştir. Bu model örüntü tanıma ve sınıflandırma gibi verideki benzerliklerin analizinde kullanılmıştır.

1982 yılında Hopfield geleneksel programlama yöntemleriyle çözülemeyen problemlerin yapay sinir ağlarıyla çözülebileceğini göstermiştir. Bu gelişme ile birlikte yapay sinir ağlarına olan ilgi artmıştır.

1982 yılında Kohonen tarafından kendi kendine öğrenme nitelik haritaları konusunda çalışmıştır. Bu sayede öğretmensiz öğrenme modelinin gelişmesine katkı sağlamıştır.

1986 yılında Rummelhart, Hinton, Williams ve McClelland tarafından geri yayılım (BP) algoritması kullanılarak çok katmalı ağların, tek katmanlı ağların çözemeyeceği problemleri çözebildiği gösterilmiştir. XOR problemi üzerinde geri yayılım algoritmasını test etmişlerdir ve çözmüşlerdir. Böylece yapay sinir ağlarının doğrusal olmayan problemlerde işe yaramadığı fikrini de yıkmışlardır.

1988 yılında Broomhead ve Lowe Radyal tabanlı Fonksiyonlar modelini geliştirmişlerdir. Bu ağlar filtreleme ve veri sıkıştırma gibi işlemlerde kullanılmıştır. Daha sonrasında Specht bu ağları geliştirmiştir ve olasılıksal sinir ağlarını ve genel regresyon ağlarını ortaya atmıştır.

90'lı yıllarda ise yapay sinir ağları ile ilgili çalışmalar yoğunluk kazanmıştır. Yapay sinir ağlarının eğitimi ile alakalı gelişmeler de olmuştur. Bu yıllarda eğitim için harcanan süreyi kısaltmak, yeni algoritmalar geliştirmek, yüksek doğrulukta sonuçlar elde etmek gibi amaçlar da yapay sinir ağlarının önemli bir konusu olmuştur.

3.3 Yapay Sinir Ağlarının Kullanım Alanları

Yapay sinir ağlarının kullanım alanları oldukça geniştir. Herhangi bir sınırı yoktur. Doğrusal olmayan, karmaşık, gürültülü, çok boyutlu, eksik verili problemlere uygulanabilmektedir. YSA, matematiksel modelin hesapsal zaman içinde çözümünün imkânsız olduğu veya matematiksel modelinin kurulamadığı problemlerde de kullanılabilir. Gerçek hayattaki kullanım alanlarına örnek olarak tıp, kalite kontrol, savunma sanayi, arıza tespiti, üretim, parmak izi tanıma, yüz tanıma, ses tanıma, robotik sistemler, akıllı sistemler, kredi verme kararı, finans, haberleşme gibi alanlar verilebilir [2].

3.4 Yapay Sinir Ağlarının Avantajları

Yapay sinir ağlarının geleneksel yöntemlere göre üstünlükleri oldukça fazladır. Özellikle geleneksel yöntemlerle çözülemeyen problemlerin yapay sinir ağlarıyla çözülmesi ve farklı algoritmalarla eğitilebilmesi oldukça önemlidir. Genel olarak yapay sinir ağlarının özellikleri aşağıda sıralanmıştır.

Doğrusal olmayan yapı: Gerçek hayattaki problemler genellikle doğrusal olmayan problemlerdir. Bu durumda doğrusal yapıdaki problem çözme teknikleri ile bu problemler ya çözülemez ya da yanlış bir çözüm elde edilir. YSA, doğrusal olmayan bir yapıdadır ve doğrusal olmayan problemlerin çözümünde kolaylıkla kullanılabilir [7].

Paralellik: Bilgi işlem yöntemleri genellikle seri işlemlerden oluşmaktadır. Seri işlemlerin birisinde bir yavaşlama olması tüm sistemi yavaşlatmaktadır. Bu istenmeyen bir durumdur. Yapay sinir ağları ise paralel işlemci kullanmaktadır. İşlemlerden birisinde oluşan yavaşlama sistemin geneline etki etmez. Bu durumda yapay sinir ağları daha hızlı çalışmaktadır [9].

Genelleme: Yapay sinir ağları kendisine öğretilen örnekler üzerinden daha önce karşılaşmadığı örnekler için genelleme yapabilme özelliğine sahiptirler. Eksik ve hatalı veriler olması durumunda bile iyi birer genelleycidirler. Bu özellikleri sayesinde örüntü tanımlama ve sınıflandırma gibi alanlarda etkilidirler [7].

Eksik verilerle çalışma: Yapay sinir ağları hem eğitim aşamasında hem de test aşamasında eksik verilerle çalışabilmektedirler. Eksik verinin durumuna ve sıklığına göre performansı değişebilmektedir [2].

Hata toleransı: Yapay sinir ağları işlemcilerinde paralel bağlantı vardır. Bundan dolayı ağın sahip olduğu bilgi tüm bağlantılara dağılmıştır. Böylece girdideki hatalı veri bütün ağırlıklara dağılacaktır. Bu sayede hatalı bir veri sistemde çok büyük hatalara sebep olmaz ve hata kolayca tolere edilebilir [9].

Öğrenme: Yapay sinir ağları insan beyni ile benzer bir yaklaşım kullanmaktadır. Bu sayede insanda bulunan öğrenme, tecrübe etme gibi özelliklere sahiptirler. Geleneksel yöntemler için zor problemler bu özellik sayesinde yapay sinir ağları için kolaydır. Ayrıca bu öğrenme işleminin sonu yoktur. Örnekler çoğaldıkça yapay sinir ağları sürekli öğrenmektedirler [7].

Çok sayıda değişken: Yapay sinir ağları sınırsız sayıda değişken ve parametre ile çalışabilmektedir. Bu sayede büyük problemlere uygulanabilmektedir [7].

Yavaş bozulma: Yapay sinir ağlarının bozulma hızları yavaştır. Hata ve eksikliklerle karşılaşıldığında hata toleransına sahip olduklarından dolayı hemen bozulmazlar [2].

3.5 Yapay Sinir Ağlarının Dezavantajları

Yapay sinir ağlarının avantajları olduğu gibi dezavantajları da vardır. Bunlardan bazıları aşağıda sıralanmıştır.

Donanımsal kısıtlar: Yapay sinir ağları donanıma bağlıdırlar. Verimli bir biçimde çalışabilmesi için hızlı ve paralel işlemcilerle ihtiyaç duymaktadırlar [2].

Probleme uygun olmama: Probleme uygun ağ yapısının belirlenmesi deneme yanılma yöntemiyle bulunmaktadır. Eğer probleme uygun bir ağ yapısı oluşturulamaz ise kötü çözümler elde edilebilir. Yapay sinir ağları kabul edilebilir çözümler üretir, optimum çözümü garanti etmez [2].

Ağ yapısı: Problem çözümünde kullanılacak olan yapay sinir ağının yapısındaki nöron sayısı, katman sayısı, eğitim algoritması gibi durumların seçimi deneme yanılmaya göre yapılır, bunları belirlemek için kesin bir kural yoktur. Bu durumda en iyi çözüm için kesin bir ağ yapısı yoktur. Oluşturulan yapının optimum olduğunun garantisi verilemez [2].

Girdilerin tanımlanması: Yapay sinir ağları sadece sayısal verilerle çalışır. Dilsel veri içeren problemlerde verinin sayısal formata çevrilmesi gerekmektedir. Bunu kullanıcı yapmak zorundadır. Eğer kullanıcı bu dönüşümü iyi yapamazsa ağın performansı da buna bağlı olarak düşebilir [2].

Eğitim problemi: Bir yapay sinir ağı eğitilirken eğitimin nerede durdurulacağı kesin olarak belli değildir. Eğitimin durdurulacağı zaman kullanıcı tarafından belirlenmektedir. Eğer eğitim erken bitirilirse ağ yeterince eğitilmemiş olur ve düşük performans gösterir. Eğer eğitim geç bitirilirse bu sefer de ağ örnekleri ezberlemiş olur ve sadece o örnekler üzerinde iyi sonuçlar verir, yeni örneklerde düşük performans gösterir [2].

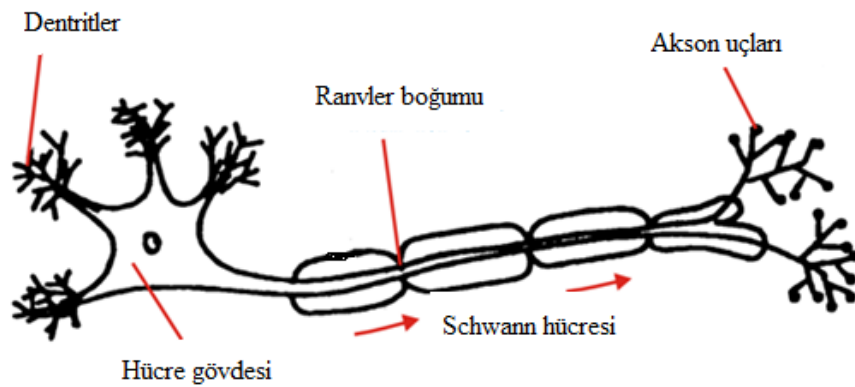
Kara kutu problemi: İstatistiksel tekniklerden alınan sonuçlar bazı testler sayesinde yorumlanabilme ve analiz edilebilme özelliğine sahiptir. Yapay sinir ağları için böyle

bir özellik yoktur. Ağın çıktıya nasıl ulaştığı gizlidir. Bilgi ağın içerisinde bağlantılarda saklıdır ve bu bilgi yorumlanabilecek bir durumda değildir [10].

3.6 Biyolojik Sinir Hücresi

Yapay sinir ağları biyolojik sinir ağından esinlenilerek geliştirilmiş bir yöntemdir. Biyolojik sinir ağları insan beyninin çalışmasını sağlayan en temel taşlardan biridir. İnsanın bütün davranışlarını ve çevresini anlamasını sağlar. Biyolojik sinir ağları beş duyu organından gelen bilgiler ışığında algılama ve anlama mekanizmalarını çalıştırarak olaylar arasındaki ilişkileri öğrenir [2].

Duyu organlarından gelen bilgiler sinir sistemi sayesinde beyne taşınır ve beynin oluşturduğu kararları da yine sinir sistemi tarafından vücudun organlarına eylem olarak gönderir. Bir sinir hücresinin yapısı Şekil 3.1'deki gibidir. Sinir hücresi hücre gövdesi, dentrit ve akson olarak adlandırılan üç kısımdan oluşur. Dentritler diğer hücrelerden bilgi toplarlar. Bütün bu bilgiler hücre gövdesinde toplanır ve değerlendirilir. Değerlendirilen bu bilgiler diğer hücrelere aktarılmak üzere aksona iletilir. Sinir hücresinde çok sayıda dentrit vardır fakat sadece bir tane akson vardır. Hücrelerin birbirleri ile olan iletişimi akson ve dentritler arasındaki bağlantı ile sağlanır. Bu bağlantı yerlerine ise sinaps denir [2].



Şekil 3.1 Biyolojik sinir hücresi

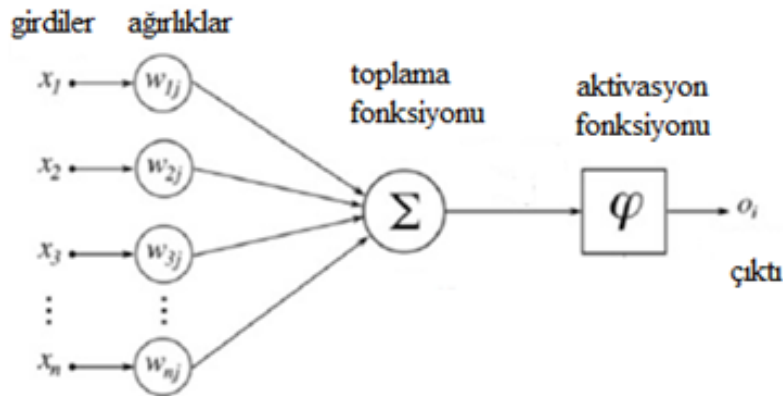
3.7 Yapay Sinir Hücresi

Biyolojik sinir ağının sinir hücrelerinden oluşması gibi, yapay sinir ağları da yapay sinir hücrelerinden oluşmaktadır. Bir yapay sinir ağı birbirleriyle bağlantılı çok sayıdaki yapay sinir hücrelerinden meydana gelmektedir. Yapay sinir hücreleri biyolojik sinir hücrelerinin basit bir modelidir. Bu modelde biyolojik sinir sistemi elemanlarının yapay sinir sisteminde karşılıkları vardır ve Tablo 3.1’de gösterildiği gibidir [11].

Tablo 3.1 Biyolojik sinir sistemi ve yapay sinir sistemindeki karşılıkları

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	Yapay sinir hücresi
Dentrit	Toplama fonksiyonu
Hücre gövdesi	Aktivasyon fonksiyonu
Akson	Çıktı
Sinaps	Bağlantı ağırlıkları

Bir yapay sinir hücresinin beş temel elemanı vardır. Bunlar; girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktıdır. Bir yapay sinir hücresinin yapısı Şekil 3.2’de gösterilmiştir.



Şekil 3.2 Yapay sinir hücresi [12]

Girdiler: Girdiler, yapay sinir hücresine dışarıdan gelen bilgilerdir. Bu bilgiler; veri kümesinin elemanları, başka bir sinir hücresinden gelen çıktılar olabileceği gibi kendinden gelen bir bilgi de olabilir.

Ağırlıklar: Ağırlıklar, sinir hücresi tarafından alınan girdilerin sinir hücre gövdesine iletilmeden önceki etkisini belirleyen değerlerdir. Ağırlıkların küçük veya büyük değerde olması o girdinin önemli veya önemsiz olduğu anlamına gelmez. Bu ağırlıklar

negatif değerler de alabilirler. Yapay sinir ağları eğitiminde algoritmalara göre ağırlık değerleri değişkenlik gösterebilir. Ağırlık değerleri değiştikçe ağırlık performansı değişir.

Toplama fonksiyonu: Toplama fonksiyonu ile hücreye gelen net girdi hesaplanır. Bu hesaplama için farklı fonksiyonlar kullanılabilir. Bu fonksiyonların en yaygın kullanılanı ağırlıklı toplam fonksiyonudur. Bu fonksiyonda her girdi değeri kendi ağırlık değeriyle çarpılarak toplanır. Bu toplam net girdiyi ifade eder. Net girdinin formülü Eşitlik (3.1)'de gösterildiği gibidir.

$$Net\ girdi = \sum_{i=1}^n G_i A_i \quad (3.1)$$

Bu denklemde G_i i . girdiyi, A_i i . girdinin ağırlığını, n ise hücreye gelen toplam girdi sayısını ifade etmektedir. Bu denklem dışında kullanılan farklı denklemler de mevcuttur:

Çarpım toplama fonksiyonu:

$$Net\ girdi = \prod_{i=1}^n G_i A_i \quad (3.2)$$

Maksimum toplama fonksiyonu:

$$Net\ girdi = Max(G_i A_i) \quad (3.3)$$

Minimum toplama fonksiyonu:

$$Net\ girdi = Min(G_i A_i) \quad (3.4)$$

Kümülatif toplam toplama fonksiyonu:

$$Net\ girdi = Net(eski) + \sum_{i=1}^n G_i A_i \quad (3.5)$$

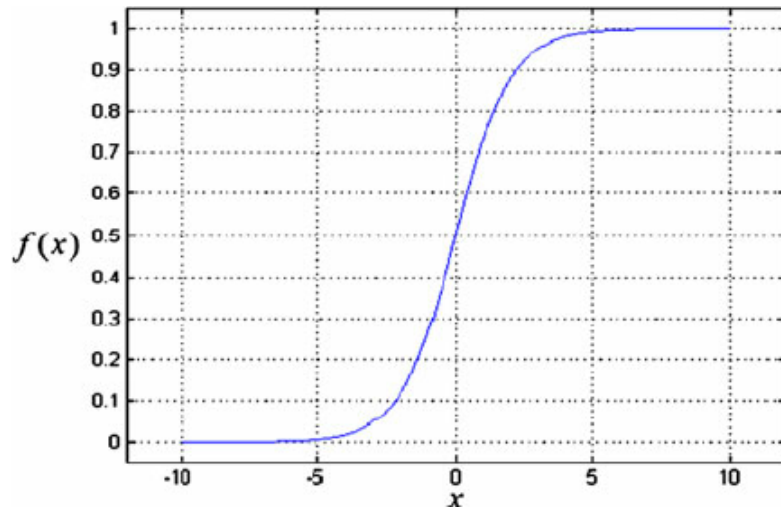
Bir problem için en iyi toplama fonksiyonunu bulmanın kesin bir yolu yoktur. Deneme yanılma yolu ile saptanabilir.

Aktivasyon fonksiyonu: Aktivasyon fonksiyonu, toplama fonksiyonundan gelen girdiyi işleyerek yapay sinir hücresinin çıkışını belirler. Transfer fonksiyonu olarak da bilinir.

Çeşitli tipleri vardır ve genellikle doğrusal olmayan bir fonksiyondur. Doğrusal olarak kullanılmamasının sebebi doğrusal fonksiyonlarda girdi ile çıktının doğru orantılı olmasıdır. Bu durum ilk yapay sinir ağları denemelerinde başarısız sonuçların alınmasına sebep olmuştur. En sık kullanılan aktivasyon fonksiyonu sigmoid ve tanjant hiperbolik aktivasyon fonksiyonlarıdır. Farklı türde aktivasyon fonksiyonları mevcuttur [2]:

Sigmoid fonksiyonu; sürekli, doğrusal olmayan ve türevi alınabilir bir fonksiyondur. En çok kullanılan fonksiyondur. Her bir girdi değeri için 0 ile 1 arasında bir çıktı değeri vermektedir. Formülü Eşitlik (3.6)'daki gibidir. Koordinat düzlemindeki görünümü de Şekil 3.3'de gösterilmiştir.

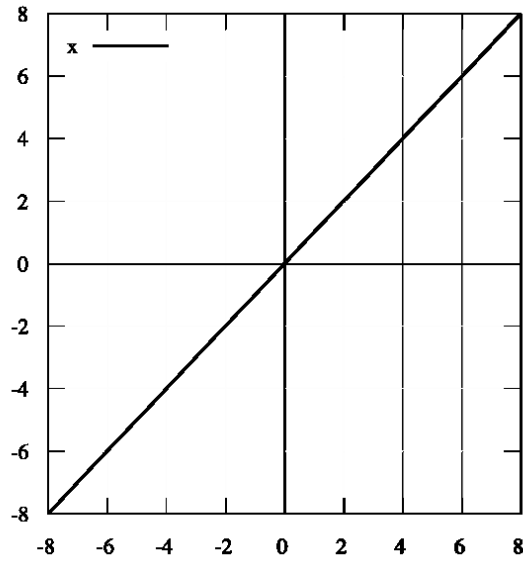
$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$



Şekil 3.3 Sigmoid fonksiyonu [13]

Doğrusal fonksiyon; sürekli, türevlenebilir ve doğrusal bir fonksiyondur. Girdi olarak gelen değeri değiştirmeden çıktı olarak verir. Formülü Eşitlik (3.7)'deki gibidir. Gösterimi ise Şekil 3.4'de verilmiştir.

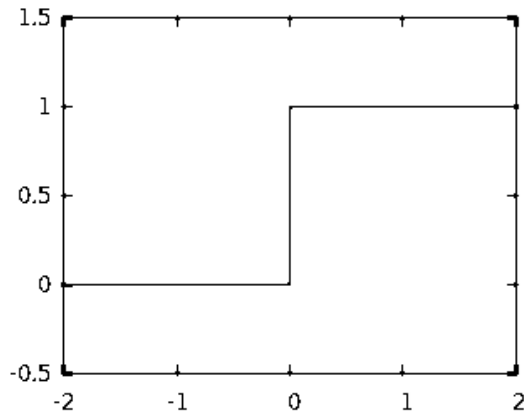
$$f(x) = x \quad (3.7)$$



Şekil 3.4 Doğrusal fonksiyon

Basamak fonksiyonu; doğrusal olmayan ve türevlenemeyen bir fonksiyondur. Girdi değerinin belirli bir basamak değerinin üstünde olup olmadığında göre değer alır. Eğer girdi değeri basamak değerinin üstündeyse 1, değilse 0 değerini alır. Formülü Eşitlik (3.8)'deki gibidir. Koordinat düzleminde gösterimi ise Şekil 3.5'de gösterilmiştir.

$$f(x) = \begin{cases} 0, & x < \text{eşik değeri} \\ 1, & x \geq \text{eşik değeri} \end{cases} \quad (3.8)$$

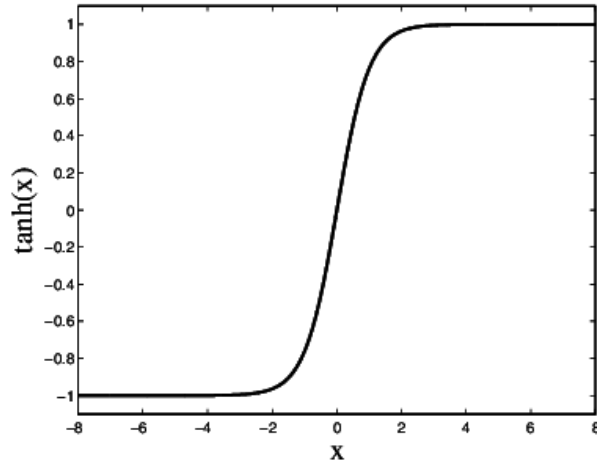


Şekil 3.5 Basamak fonksiyonu

Hiperbolik tanjant fonksiyonu; sürekli, türevlenebilir ve doğrusal olmayan bir fonksiyondur. Sigmoid fonksiyonuna benzer. Sigmoid fonksiyonu 0 ile 1 arasında değer alırken hiperbolik tanjant fonksiyonu -1 ile 1 arasında değer alır. Bu yönüyle sigmoid

fonksiyonundan farklıdır. Formülü Eşitlik (3.9)'daki gibidir. Gösterimi ise Şekil 3.6'de gösterilmiştir. Şekil olarak da sigmoid fonksiyonuna benzemektedir.

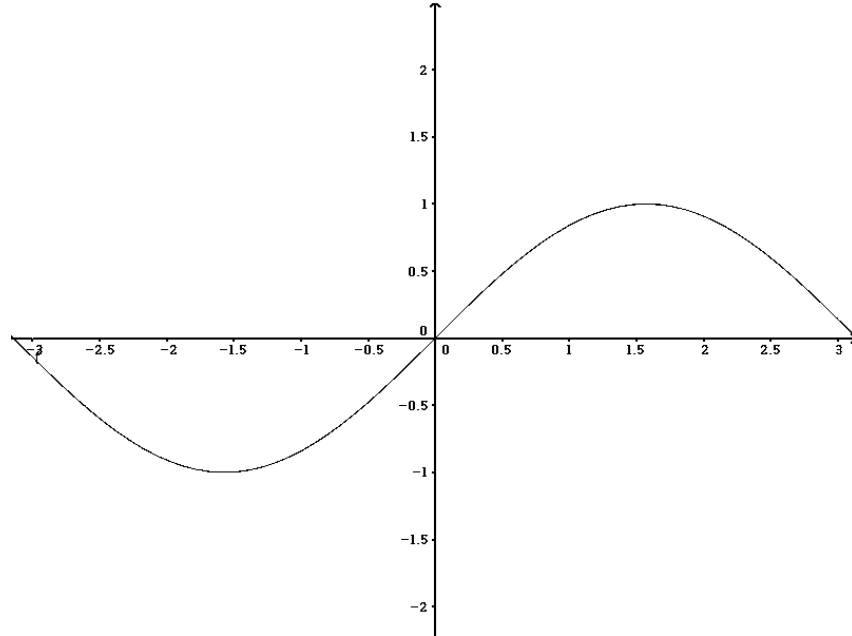
$$f(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}} \quad (3.9)$$



Şekil 3.6 Hiperbolik tanjant fonksiyonu [14]

Sinüs fonksiyonu; sürekli, türevlenebilir ve doğrusal olmayan bir fonksiyondur. Formülü Eşitlik (3.10)'daki gibidir. Şekil olarak Şekil 3.7'de gösterilmiştir.

$$f(x) = \sin(x) \quad (3.10)$$

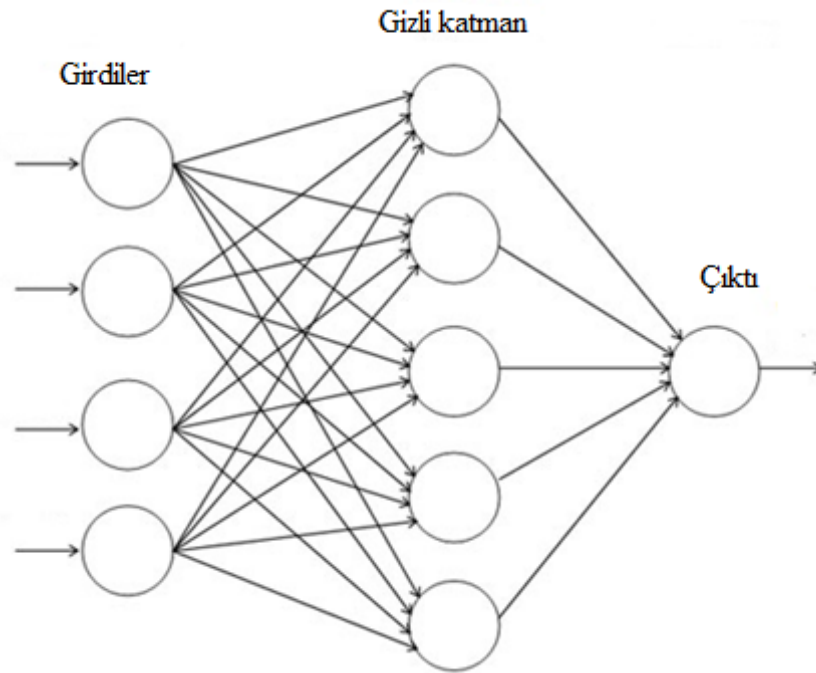


Şekil 3.7 Sinüs fonksiyonu

Çıktı: Çıktı değeri, yapay sinir hücresinin verdiği sonuçtur. Diğer bir hücreye girdi olarak gönderilebileceği gibi kendi kendine de gönderilebilir.

3.8 Yapay Sinir Ağlarının Yapısı

Yapay sinir hücreleri çok sayıda bir araya gelip, birbirleriyle bağlantılı şekilde çalıştıkları zaman yapay sinir ağını oluştururlar. Bir ağda çok sayıda hücre bulunur. Bu hücrelerin bir kısmı aynı doğrultu üzerinde bulunurlar. Bu aynı doğrultu üzerindeki hücreler topluluğuna katman denir. Temel olarak YSA'da üç katman vardır. Bunlar girdi katmanı, gizli katman ve çıktı katmanıdır. Bir yapay sinir ağının yapısı ve katmanlar şekilsel olarak Şekil 3.8'de gösterilmiştir.



Şekil 3.8 Yapay sinir ağlarının yapısı

Girdi katmanı: Dış dünyadan gelen bilgileri alırlar ve gizli katmana iletirler. Girdileri hücrelerin dizayn edilişine göre birebir aynı şekilde veya farklılaştırarak gönderebilirler.

Gizli katman: Tek bir katmandan da oluşabilir, birden fazla katmandan da oluşabilir. Girdi katmanından gelen bilgiyi işler ve çıktı katmanına iletir.

Çıktı katmanı: Çıktı katmanı, gizli katmandan gelen bilgileri işler ve yapay sinir ağının genel çıktısını verir. Çıktı katmanında tek hücre olabileceği gibi birden fazla hücre de olabilir. Problemin türüne göre çıktı değerlendirilir.

3.9 Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları farklı şekillerde sınıflandırılabilir. Ağlar içerdikleri katman sayılarına göre, hücrelerin birbirleri ile olan bağlantılarına göre ve bu bağlantılardaki değerlerin değiştirilerek ağın nasıl eğitildiğine göre sınıflandırılabilirler.

3.9.1 Katman Sayısına Göre Sınıflandırma

Yapay sinir ağları, çok sayıdaki sinir hücrelerinin birbirleriyle bağlantı kurmasıyla oluşur. Bu hücreler aynı doğrultuda bulunarak katmanları oluştururlar. Yapay sinir ağlarında girdi ve çıktı katmanları bulunur, bazılarında bunlara ilaveten gizli katman da bulunur. Gizli katman tek katmandan oluşabileceği gibi çok sayıda katmandan da oluşabilir.

Tek Katmanlı Yapay Sinir Ağları: Girdi ve çıktı katmanlarından oluşur. Gizli katman yoktur. Doğrusal problemlerin çözümünde iyi sonuç verir. Doğrusal olmayan problemlerde iyi sonuç vermez. İlk geliştirilen yapay sinir ağları tek katmanlıdır.

Çok Katmanlı Yapay Sinir Ağları: Girdi ve çıktı katmanlarının yanında gizli katmana da sahip olan yapay sinir ağıdır. Hem doğrusal problemlerde hem de doğrusal olmayan problemlerde iyi sonuçlar verir. Tek katmanlı ağlara göre eğitilmeleri daha zordur.

3.9.2 Yapılarına Göre Sınıflandırma

Yapay sinir ağları yapılarına göre ileri ve geri beslemeli ağlar olmak üzere iki gruba ayrılabilir.

İleri Beslemeli Ağlar: İleri beslemeli ağlarda, hücreler kendilerinden sonraki katmandaki hücrelerle bağlantı kurarlar. Kendi katmanındaki hücreler ile bağlantı kurmazlar. Bilgi aktarımı girdi katmanından gizli katman, gizli katmandan ise çıktı katmanına doğru olur. Geriye doğru bir bilgi akışı yoktur.

Geri Beslemeli Ağlar: Geri beslemeli yapay sinir ağlarında hücreler arasında geriye doğru bağlantılar veya kendilerine doğru bağlantılar bulunmaktadır.

İleri beslemeli ağlar geri beslemeli ağlara göre daha çok kullanılmaktadır. Bunun sebebi geri beslemeli ağların pratik olarak kullanımının zor olmasıdır. Geri beslemeli ağların eğitiminde tutarsızlıklar çok görülür [7].

3.9.3 Öğrenme Yöntemine Göre Sınıflandırma

Yapay sinir ağları öğrenme yöntemlerine göre üçe ayrılabilir. Bunlar danışmanlı öğrenme, danışmansız öğrenme ve destekli öğrenmedir.

Danışmanlı Öğrenme: Danışmanlı öğrenmede, yapay sinir ağına öğrenme için girdi değerlerinden ve hedef çıktı değerlerinden oluşan örnek bir veri kümesi verilir. Verilen bu hedef çıktı değerleri danışman olarak adlandırılır. Hedeflenen çıktı değerleri ile ağı verdiği çıktı değerleri arasında farklılıklar vardır. Bu farklılıklar üzerinden ağı performansı ölçülür. Bunun için bazı performans ölçütleri vardır. Genellikle hataların karelerinin toplanıp ortalamasının alınmasıyla elde edilen değerin karekökü alınır ve bu değer performans ölçütü olarak kullanılır. Ağ eğitildikçe bu performans ölçütlerinin değeri değişir ve ağ daha iyi bir hale gelir [15].

Danışmansız Öğrenme: Danışmansız öğrenmede, ağa verilen örnek veriler sadece girdilerden oluşur. Hedeflenen çıktı değerini içermez. Dolayısıyla ağı çıktısı ile hedeflenen çıktıyı karşılaştırma gibi bir imkân yoktur [15].

Destekli Öğrenme: Destekli öğrenmede, yapay sinir ağına bir danışman yardımcı olur. Her girdi seti için üretilmesi gereken çıktı değerini danışman sisteme göstermez. Sistem girdilere göre çıktı üretir, danışman ise çıktının doğru veya yanlış olduğunu belirtir. Bu gelen bildirimlere göre ağ öğrenmesini devam ettirir [2].

3.10 Öğrenme Kuralları

Yapay sinir ağlarının öğrenmesi için başlangıçta rastgele olarak atanan ağırlık değerlerinin değiştirilmesi gerekmektedir. Bu değerler değiştikçe ağ eğitilecek ve daha iyi çıktılar verecektir. Ağırlık değerlerinin değiştirilmesi kullanılan öğrenme stratejisine

ve kurallarına göre şekillenir. Öğrenme kuralları çevrimiçi ve çevrimdışı olarak ikiye ayrılabilir.

Çevrimiçi Öğrenme: Çevrimiçi öğrenmede gerçek zamanlı öğrenme vardır. Ağ hem öğrenirken hem de çalışıp çıktı vermektedir. Bundan dolayı yeni veriler eklendiği durumlarda tüm sistemi en baştan eğitmeye gerek kalmaz [2].

Çevrimdışı Öğrenme: Çevrimdışı öğrenmede sistem önce belirli örnekler üzerinden eğitilir. Eğitim bitirilip sistem kullanılmaya başlandığında, ağ artık öğrenmemektedir. Eğer sistemin eğitimi için yeni veriler eklenmesi gerekirse sistem kapatılıp en baştan eğitim işleminin tekrarlanması gerekmektedir. Eğitim işlemi tekrarlandığında sistem yeniden işleme sokulmaktadır [2].

Çok sayıda öğrenme kuralı geliştirilmiştir. Temel olarak en çok bilinen kullanılanları Hebb, Hopfield, Kohonen ve Delta öğrenme kurallarıdır.

Hebb Kuralı: Hebb tarafından geliştirilen bir kuraldır. İlk öğrenme kuralıdır. Bu kurala göre eğer bir sinir hücresi başka bir sinir hücresinden girdi aldığında, bu iki hücre aynı işarete sahipse hücreler arasındaki ağırlık güçlendirilmelidir. Bir başka deyişle eğer bir sinir hücresi aktif ise bağlı olduğu hücreleri de aktif yapmaya, pasif ise bağlı olduğu hücreleri pasif yapmaya çalışır [2, 16].

Hopfield Kuralı: Hopfield kuralı, Hebb kuralına benzemektedir, fakat Hebb kuralından bir farklılığı vardır. Bu farklılık ağırlıkların güçlendirilmesi işleminde bir büyüklük belirlemek gerekmesi olarak ifade edilebilir [16].

Kohonen Kuralı: Kohonen kuralına göre yapay sinir ağının hücreleri ağırlıklarını değiştirmek için birbirleriyle yarışır. En büyük çıktıyı veren hücre kazanır ve ağırlıkları değiştirilir. Bu sayede hem o hücre hem de komşularının ağırlıkları değişir ve o hücre daha kuvvetli hale gelir [16].

Delta Kuralı: Delta kuralı Hebb kuralının geliştirilmiş bir halidir. Hedeflenen çıktı değeri ile ağdan elde edilen çıktı değeri arasındaki farkın azaltılması hedeflenmektedir. Bunun için bağlantı değerleri sürekli değiştirilir ve en iyi sonuca ulaşılmaya çalışılır [16].

3.11 Yapay Sinir Ağı Tasarımı

Bir yapay sinir ağının iyi sonuçlar verebilmesi için ağın iyi tasarlanmış olması gerekmektedir. En verimli yöntemlerin ve durumların seçilmesi önemlidir. Bunun kesin bir kuralı yoktur, deneme yanılma metoduyla saptanabilir. Bir yapay sinir ağının tasarımında ağ mimarisinin seçilmesi, öğrenme algoritmasının seçilmesi, gizli katman sayısının belirlenmesi, gizli katmandaki eleman sayısı gibi durumlarla karşılaşılmaktadır.

Ağ Yapısının Seçimi: Yapay sinir ağlarında probleme göre ağ yapısı farklılık gösterebilir. Problemin amacına göre uygun bir ağ yapısı seçilmelidir. Sınıflandırma, tahmin, veri ilişkilendirme gibi farklı amaçlar için farklı ağ yapıları kurulmalıdır. Yanlış kurulan bir ağ yapısı performans, doğruluk ve modelleme yeteneği açısından olumsuz sonuçlar verebilir.

Öğrenme Algoritmasının Seçimi: Bir yapay sinir ağının eğitimi için kullanılacak eğitim algoritmasının seçimi çok önemlidir. Bunun için çok sayıda eğitim algoritması mevcuttur. En çok kullanılan algoritma geri yayılım algoritmasıdır. Bunun dışında metasezgisel algoritmalar da yaygın şekilde kullanılmaktadır. Bir yapay sinir ağının eğitiminde iyi sonuçlar veren bir algoritma başka bir yapay sinir ağının eğitiminde kötü sonuçlar verebilir. Aynı şekilde bir ağda kötü sonuç veren bir algoritma ise başka bir ağda iyi sonuçlar verebilir.

Gizli Katman Sayısının Belirlenmesi: Katmanlar sinir hücrelerinin aynı doğrultuda bulunmasıyla oluşur. En ideal katman sayısını bulmanın en uygun yöntemi deneme yanılma yapmaktan geçer. Gizli katman sayısının çok olması yapay sinir ağını daha karmaşık bir hale getirir ve işlem süresini uzatır. Gizli katmanın çok olması her zaman olumlu etki yapmayabilir.

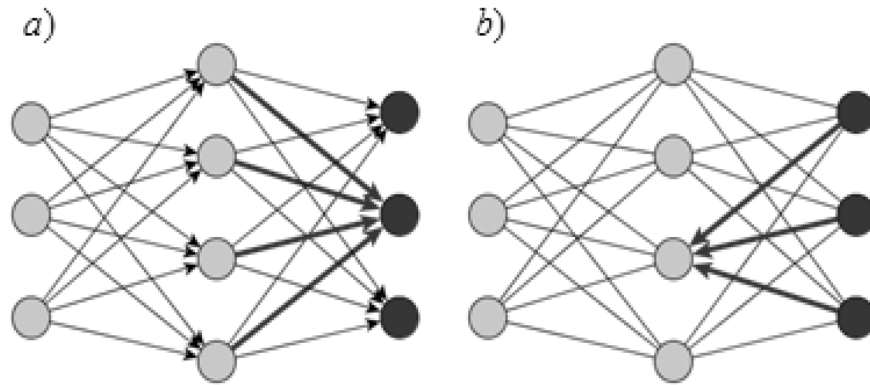
Nöron Sayısının Belirlenmesi: Tıpkı gizli katman sayısının bulunması için deneme yanılma yapılması gibi ideal nöron sayısını bulmak için de deneme yanılma yapmak gerekmektedir. İdeal bir sayısı yoktur. Nöron sayısının az olması yapay sinir ağının genelleme yapma kabiliyetini artırır. Çok olması ise verilerin ezberlenmesine yol açar.

Fonksiyonların Belirlenmesi: Yapay sinir ağı tasarlanırken ağda kullanılan bazı fonksiyonlar vardır. Aktivasyon fonksiyonu, toplama fonksiyonu bunlara örnek olarak verilebilir. Genellikle toplama fonksiyonu olarak girdinin ağırlıklarla çarpılmış hali kullanılır, aktivasyon fonksiyonu olarak ise sigmoid fonksiyonu kullanılır. Bunun gibi farklı fonksiyonları denemek ağın performansını değiştirebilir.

Eğitim ve Test Verilerinin Belirlenmesi: Yapay sinir ağları çalışırken girdi olarak verileri kullanılır. Bu verilerin bir kısmı eğitim için diğer kısmı ise test için kullanılır. Bunun dışında bütün veriler yöntemine göre test ve eğitim verisi olarak da kullanılabilir. Eğitim ve test verilerinin seçimi ağın performansını etkileyen bir faktördür.

3.12 Geri Yayılım Algoritması

Yapay sinir ağı eğitiminde en çok kullanılan algoritmadır. Hataları çıktı katmanından girdi katmanına geriye doğru yönlendirdiği için geri yayılım ismini almıştır. Geri yayılım algoritması ileri besleme ve geri besleme olarak iki kısma ayrılabilir ve bu kısımlar Şekil 3.9'daki gibi ifade edilebilir.



Şekil 3.9 a) İleri besleme b) Geri besleme [17]

Çok katmanlı bir ileri beslemeli ağ için algoritma aşağıdaki gibidir [1].

Algoritmada;

q : katman numarası

H_i^q : q 'inci katmandaki i biriminin girdisi

y_i^q : q 'inci katmandaki i biriminin çıktısı

δ_i^q : q 'inci katmandaki i birimindeki hata miktarı

v_i^q : q 'inci katmandaki i birimindeki sonuç

n : öğrenme katsayısı

w_{ij}^q : $(q-1)$ 'inci katmandaki i birimini, q 'inci katmandaki j birimine bağlayan ağırlık değerini ifade etmektedir.

Algoritmanın adımları aşağıdaki gibidir.

1. w 'ye başlangıç olarak rasgele sayılar atanır.
2. Katmanlardaki her bir eleman için ileri yöndeki çıktı değeri Eşitlik (3.11) ile hesaplanır.

$$y_i^q = f\left(\sum_i y_i^{q-1} \times w_i^{q-1}\right) \quad (3.11)$$

3. Çıkış birimleri için hata miktarları hesaplanır.

$$\delta_i^Q = (v_i^Q - y_i^Q) \times f'(H_i^Q) \quad (3.12)$$

4. $Q=2, 3, \dots, Q-1$, Q katmanlarındaki bütün i birimleri için geriye yayılımla deltalar yani gizli katman birimleri için hata terimleri hesaplanır.

$$\delta_i^{q-1} = f'(H_i^{q-1}) \times \sum_i (\delta_i^q \times w_i^q) \quad (3.13)$$

5. Bütün ağırlıklar w_{ij} 'ler kullanılarak güncellenir.

$$w_{ij}^{yeni} = w_{ij}^{eski} + \Delta w_{ij}^q \quad (3.14)$$

$$\Delta w_{ij}^q = n \times \delta_i^q \times y_i^{q-1} \quad (3.15)$$

6. 2. Adıma dönülür ve toplam hata kabul edilebilir bir düzeye gelene kadar işlemler tekrarlanır.

Geri yayılım algoritmasının çalışması için transfer fonksiyonunun türevlenebilir bir fonksiyon olması gerekmektedir. Bunun için genellikle sigmoid fonksiyonu kullanılır. Sigmoid fonksiyonu Eşitlik (3.16)'daki gibidir, türevi ise Eşitlik (3.17)'deki gibi olmalıdır.

$$f(x) = \frac{1}{1 + e^{-\beta x}} \quad (3.16)$$

$$f'(x) = \beta \times f(x) \times (1 - f(x)) \quad (3.17)$$

4. BÖLÜM

YAPAY SİNİR AĞLARI EĞİTİMİ LİTERATÜR TARAMASI

Yapay sinir ağları eğitiminde kullanılan çeşitli algoritmalar vardır. En çok kullanılanı geri yayılım algoritmasıdır. Bunun dışında eğitimde metasezgisel algoritmalar da kullanılmıştır. Bunun sebebi metasezgisel algoritmaların hızlı çalışabilmesi ve yerel minimum noktaya takılmama gibi avantajlarının olmasıdır. Geçmişte yapay sinir ağlarının eğitimi ile alakalı birçok çalışma yapılmıştır.

Sexton ve Gupta 2000 yılında yaptıkları çalışmada genetik algoritma (GA) ile geri yayılım algoritmasını yapay sinir ağlarının eğitiminde karşılaştırmışlardır [18]. Her iki algortmada da gizli katmandaki nöron sayıları değiştirilerek denenmiştir. Gizli katmandaki nöron sayısını artırmak her zaman iyi sonuç vermemiştir. Amaç fonksiyonu olarak karesel hatalar toplamının minimizasyonu kullanılmıştır. Bütün testlerde GA BP algoritmasından daha iyi sonuçlar vermiştir. Ayrıca süre olarak da büyük üstünlük sağlamıştır.

Gudise ve Venayagamoorthy 2003 yılında yaptıkları çalışmada parçacık sürü optimizasyonu (PSO) algoritması ile geri yayılım algoritmasını karşılaştırmışlardır [19]. Bunun için lineer olmayan bir fonksiyon üzerinden ağ eğitilmiştir ve testler yapılmıştır. Bu testler sonucunda her iki algoritmanın da hata oranları karşılaştırılmıştır. PSO algoritmasının hata oranının daha düşük olduğu gözlemlenmiştir. Ayrıca altı kat daha az iterasyonda bunu başarmıştır. Bunun dışında nöronlara eşik değerinin konulup konulmaması PSO algoritmasını çok etkilememektedir, fakat BP algoritmasının performansını çok değiştirmektedir.

Blum ve Socha 2005 yılındaki çalışmalarında yapay sinir ağları eğitiminde karınca koloni optimizasyonu algoritması kullanımını ele almışlardır [20]. Karınca koloni

algoritması orijinal olarak kesikli problemler için bulunmuş olsa da sürekli problemlere de uygulanabilir. Karınca koloni algoritmasının ağ eğitiminde sağlık verileri kullanılmıştır. Bunlar Cancer, Diabetes ve Heart veri kümeleridir. Karşılaştırma için ise BP ve Levenberg-Marquardt (LM) algoritmaları seçilmiştir. Ayrıca karınca koloni algoritması ile diğerlerinin hibritleştirilmiş hali de denenmiştir. Amaç fonksiyonu olarak karesel hataların minimizasyonu belirlenmiştir. Algoritmaların performansları karşılaştırıldığında karınca koloni algoritmasının hibritleştirilmiş hallerinin diğerlerine göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Kim ve arkadaşlarının 2005 yılında yaptıkları çalışmada genetik algoritmanın geliştirilmiş bir hali ile yapay sinir ağı eğitilmiştir [21]. Amaç fonksiyonu olarak karesel hatalar toplamı ele alınmıştır. Geliştirilen algoritma %30'a yakın oranda daha hızlı sonuçlar elde edilmiştir ve bu da kayda değer bir farklılıktır.

Dengiz ve arkadaşları 2009 yılında yapay sinir ağları eğitimi için bir tabu arama algoritması sunmuşlardır [22]. Geleneksel yöntemler olan BP, LM gibi algoritmalar türev tabanlı algoritmalar. Çok popüler olmalarına rağmen iki dezavantajları vardır. Yerel optimum noktaya takılıp kalırlar ve hızları düşüktür. Bir yapay sinir ağının eğitimi doğrusal olmayan bir optimizasyon yöntemidir. Ağı eğitmek için tabu arama algoritmasını temel alan yeni bir algoritma geliştirilmiştir. Bu geliştirilen algoritma tabu arama, genetik ve tavlama benzetimi algoritmaları ile karşılaştırılmıştır. Test problemlerinin genelinde bu yeni algoritma diğerlerinden iyi sonuçlar vermiştir ve standart sapması düşük çıkmıştır. Ayrıca kayıt tutarak ilerlediği için hız olarak da daha iyi çıkmıştır.

Kattan ve arkadaşlarının 2010 yılında yaptıkları çalışmada yapay sinir ağları eğitiminde harmoni arama (HA) algoritması kullanmışlardır [23]. Birçok stokastik global optimizasyon metodu yapay sinir ağlarının eğitimi için kullanılmıştır. Bunların çoğu genetik algoritma, karınca koloni algoritması gibi doğadan etkilenen algoritmalar. Bu algoritmalar BP algoritmasının yerel optimuma takılma sorununa sahip değildir. HA algoritması ise müzikten etkilenilerek geliştirilmiş bir metasezgisel arama algoritmasıdır ve basit problemlerde geri yayılım algoritmasından daha iyi sonuçlar verdiği gözlemlenmiştir. Bir müzik bandosundaki müzik yapma işleminden etkilenilerek geliştirilmiştir. Her müzisyen kendi müzik aletiyle bir nota çalar ve diğer müzisyenlerin

çaldıklarıyla uyumlu olup olmamasıyla müzik geliştirilir. Algoritmanın test edilmesi için Cancer veri kümesi üzerinde çalışılmıştır. Geri yayılım algoritması ile sonuçlar karşılaştırıldığında ise harmoni arama algoritmasının daha iyi olduğu görülmüştür.

Zamani ve Sadeghian 2010 yılında yaptığı çalışmada yapay sinir ağlarının eğitiminde parçacık sürü algoritmasını kullanmışlardır [24]. Veri kümeleri olarak literatürdeki Iris, Wine, Heart ve Ionosphere veri kümelerinden yararlanmışlardır. Bu veriler üzerinde çok sayıda test yapılmış ve ağlar eğitilmiştir. Eğitimler sonucunda yüksek doğruluk oranları elde edilmiştir.

Kulluk'un 2012 yılında yaptığı çalışmada harmoni ve av arama (HS) algoritmasının hibrit hali kullanılmıştır [25]. Harmoni arama algoritması müzisyenlerin müzik geliştirmesinden etkilenilerek geliştirilen bir algoritmadır. Av arama algoritması ise hayvanların grup şeklinde avlanmalarından esinlenmiştir. Algoritmalar iki aşamada test edilmiştir. İlk aşamada bu hibrit algoritma HA algoritması ve HS algoritması ile karşılaştırılmıştır. İkinci aşamada ise geleneksel algoritmalar ve metasezgisel algoritmalar ile karşılaştırılmıştır. Yapay sinir ağı eğitimindeki ana amaç hatayı minimuma indirmektir. Bunun için minimum karesel hata fonksiyonu amaç fonksiyonu olarak tanımlanmıştır. Algoritmanın test edilip diğer algoritmalarla karşılaştırılması için literatürdeki dokuz veri kümesi kullanılmıştır. Yapılan testler sonunda hibrit algoritmanın, harmoni ve av arama algoritmalarından daha iyi sonuçlar verdiği gözlemlenmiştir. Önerilen algoritma, geleneksel ve metasezgisel algoritmalarla karşılaştırıldığında ise genel olarak daha etkili sonuçlar vermiştir.

Green ve arkadaşlarının 2012 yılındaki yaptıkları çalışmada yapay sinir ağı eğitiminde merkezi kuvvet optimizasyonu algoritmasının performansını PSO algoritması ile karşılaştırmıştır [26]. Merkezi kuvvet algoritması yeni nesil metasezgisel algoritmalarından birisidir ve kütle çekim kuvvetinden ilham almıştır. Parçacık sürü optimizasyonu da popülasyon temelli stokastik bir arama algoritmasıdır. Sürü şeklindeki kuşlardan ve balıklardan esinlenmiştir. Daha önceden bu algoritma ile yapılan birçok yapay sinir ağı eğitimi vardır. İki algoritmanın performansını karşılaştırmak için farklı ağlar üzerinde test yapılmıştır. Bunlardan ilki XOR (ya da) problemidir. İkincisi iris veri kümesi için oluşturulan yapay sinir ağıdır. Testler sonucunda merkezi kuvvet algoritmasının iyi sonuçlar verdiği belirlenmiştir. İki

algoritma arasındaki en büyük fark rasgelelikleri ve arama uzayındaki davranışlarıdır. Merkezi kuvvet algoritmasının çalışma süresi biraz daha uzundur. Merkezi kuvvet algoritması diğer algoritmalarla hibritleştirildiğinde çok daha iyi ve hızlı sonuçların alınabileceği vurgulanmıştır.

Nandy ve arkadaşlarının 2012 yılındaki yaptıkları çalışmada ateşböceği algoritması ile BP algoritmasının bir hibriti olan yeni bir algoritma oluşturulmuştur [27]. Önerilen yeni algoritma literatürdeki Iris, Wine veri kümeleriyle denenmiştir. Performans ölçütü olarak karesel hata toplamı fonksiyonu ve doğruluk oranları kullanılmıştır. Çıkan sonuçlar genetik algoritma performansı ile karşılaştırılmıştır. Önerilen algoritma, performans olarak daha az iterasyon gerektirmektedir, daha hızlı çalışmaktadır ve daha az hafıza alanı kullanmaktadır.

Nandy ve arkadaşlarının 2012 yılında yaptıkları çalışmada yapay arı kolonisi (ABC) algoritması ile yapay sinir ağları eğitilmiştir [28]. ABC algoritması üzerinde bazı değişiklikler yapılarak yapay sinir ağlarına uyarlanmıştır ve çözüm için ağırlıkları ve eşik değerleri değiştirilmiştir. Performans ölçütleri olarak minimum karesel hata toplamı, doğruluk oranları, hız ve sapma oranları kullanılmıştır. Bu yeni algoritma literatürdeki Iris, Wine, Soybean ve Glass veri kümeleri üzerinden GA ile karşılaştırılmıştır. Sonuç olarak testlerde büyük üstünlük sağlamıştır.

Askarzadeh ve Rezazadeh'in 2013 yılında yaptığı çalışmada son zamanlarda bazı optimizasyon problemlerine uyarlanan ve iyi sonuçlar veren kuş eşleşme algoritması yapay sinir ağlarının eğitiminde kullanılmıştır [29]. Bu algoritmanın yeterliliğini test etmek için literatürde yer alan üç adet gerçek hayat verisi kullanılmıştır. Bunlardan Cancer ve Diabetes veri kümeleri eksik bilgiler içeren ve gürültülü veri kümeleridir. Sinir ağını eğitmek için üç katmanlı bir ağ tasarlanmıştır. Girdi katmanı ile gizli katman, gizli katman ile çıktı katmanı arasında ağırlıklar tanımlanmıştır. Ayrıca gizli katman ve çıktı katmanına girdiden bağımsız olarak ayrıca bir değer tanımlanmıştır. Deneysel tasarımlar sonucunda kuş eşleme algoritması en iyi sonucu vermese de diğerlerine göre kabul edilebilir sonuçlar vermiştir.

Yaghini ve arkadaşlarının 2013 yılında yaptığı çalışmada parçacık sürü algoritması ile geri yayılım algoritmasının hibrit hali kullanılmıştır [30]. Sekiz adet literatür verisiyle

algoritma test edilmiştir. Algoritma Java programlama diliyle kodlanmış ve performansı test edilmiştir. Problemlerin genelinde önerilen algoritma, hızlı ve etkili sonuçlar vermiştir.

5. BÖLÜM

METASEZGİSEL ALGORİTMALAR

Meta “öte, üst düzey” anlamına gelmektedir [31]. Metasezgisel terimi ilk olarak Glover (1986) tarafından kullanılmıştır. Metasezgisel, optimizasyon problemlerinin çözümü için özelleştirilebilen yüksek düzey bir algoritmik yapı veya yaklaşım; uygun veya optimal çözüm aramada diğer sezgiselleri yönlendiren yüksek düzeyli genel bir strateji veya mevcut çözümler kümesini değiştirerek iyi çözümlere bakan uzay araması olarak tanımlanabilir.

Metasezgisel algoritmaların özellikleri olarak şunlar sıralanabilir;

- Genel amaçlıdır fakat özel problemlere uyarlanabilirler.
- Yerel optimumdan kaçmayı amaçlarlar (arama uzayını ararken iyi çözümler bulurlar)
- Genel olarak NP-tam veya NP-zor olarak sınıflandırılan problemlere uygulanırlar.
- Ayrıca polinomsal-zamanda çözümün olduğu fakat pratik olmadığı diğer kombinatoriyal problemlere de uygulanırlar.
- Optimum çözümü garanti etmezler.

Metasezgiseller genel olarak,

- Doğadan esinlenen (GA) – Doğadan esinlenmeyen (TS)
- Popülasyon tabanlı – tek nokta arama (yol/rota –trajectory - metotları)
- Dinamik - statik amaç fonksiyonu
- Tekli – çoklu komşuluk yapıları
- Hafıza kullanan – düşük hafıza kullanan metotlar

şeklinde sınıflandırılabilirler.

Bu tez çalışması kapsamında popülasyon tabanlı yeni nesil 3 metasezgisel algoritma yapay sinir ağlarının eğitiminde ele alınmıştır. Bu algoritmalar;

- İyon hareketi algoritması (IM)
- Sosyal örümcek algoritması (SS)
- Stokastik fraktal arama (SFS) algoritmasıdır.

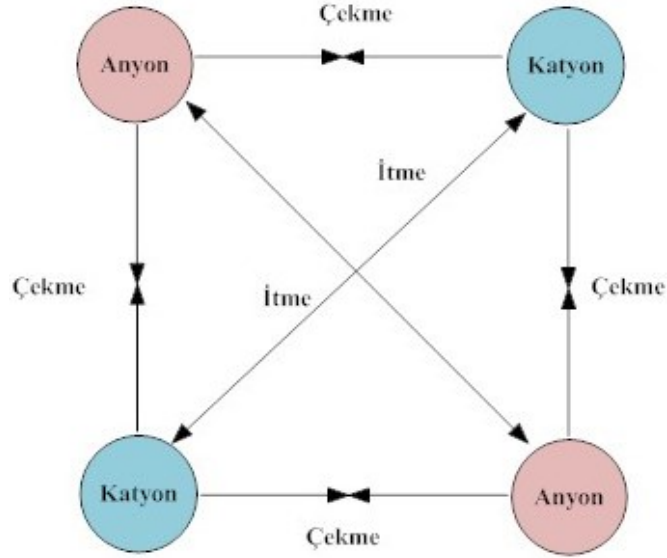
5.1 İyon Hareketi Algoritması

İyon hareketi algoritması, Javidy ve arkadaşları tarafından 2015 yılında ortaya atılan, iyonların davranışlarından etkilenilerek geliştirilmiş, popülasyon tabanlı yeni nesil bir metasezgisel algoritmadır [32]. Algoritma tasarlanırken düşük işlem yükü, hızlı çalışma ve yerel optimum noktalardan kaçınmak hedeflenmiştir. Algoritmada iyonlar, anyonlar ve katyonlar olmak üzere iki gruba ayrılmıştır. Anyonlar negatif yüklü, katyonlar ise pozitif yüklü iyonlardır. IM algoritması, aday çözümler popülasyonunu, negatif yüklü ve pozitif yüklü iyonlar olmak üzere iki kümeye ayırır ve bu çözümleri iyonların “ aynı yüklü iyonlar birbirini iterler, zıt yüklü iyonlar birbirlerini çekerler” şeklinde tanımlanan önemli özelliğine göre iyileştirirler. İyonlar arasındaki bu itme çekme özelliği Şekil 5.1’de gösterilmiştir. Algoritmanın başlangıcında veya bazı olasılıklarla iterasyonlar boyunca çözümler $r(ubi-lbi)+lbi$ fonksiyonu ile üretilir. Bu fonksiyonda r , düzgün dağılımdan $[0, 1]$ aralığında üretilen rastsal bir sayı, lbi ve ubi ise sırasıyla i . değişkenin alt ve üst sınırıdır.

IM algoritmasında her bir iyon, aday bir çözümü ifade eder. İyonlar arasında bir çekim kuvveti vardır. Bu çekim kuvvetine göre iyonlar arama uzayında hareket eder ve çözümler güncellenir. Bu hareket iyonların ters işaretli en iyi iyona doğru ilerlemesi şeklindedir. Yani anyonlar en iyi katyona, katyonlar ise en iyi anyona doğru hareket ederler. En iyi anyon ve katyon ise çözümün verdiği amaç fonksiyonuna göre en iyi durumda bulunan iyondur.

İyon hareketi algoritmasında, iyonların hareketi iterasyonlar boyunca tüm iyonların iyileşmesini garanti eder. Ancak iyonların farklılaştırılması ve kuvvetlendirilmesi için herhangi bir mekanizma yoktur. Algoritmada iyonların farklılaştırılması ve

kuvvetlendirilebilmesi için iyonların sıvı ve katı faz olmak üzere iki farklı fazda bulunabileceği varsayılmıştır.



Şekil 5.1 Anyon ve katyonların hareketi

5.1.1 Sıvı Faz (Farklılaştırma)

İyonlar sıvı fazda daha çok hareket özgürlüğüne sahiptir. Ayrıca, zıt yüklü iyonlar arasındaki çekim kuvveti, aynı yüklü iyonlar arasındaki itme kuvvetinden daha fazladır. İyon hareketi algoritmasında bu fazda arama uzayını arayabilmek için itme kuvvetleri ihmal edilmiştir. Sadece çekim kuvvetleri kullanılmıştır ve çekim kuvvetini etkileyen tek faktörün iyonlar arasındaki uzaklık olduğu varsayılmıştır. Çekim kuvvetleri ise Eşitlik (5.1) ve (5.2)'deki gibi modellenmiştir

$$AF_{ij} = \frac{1}{1 + e^{-0.1/AD_{ij}}} \quad (5.1)$$

$$CF_{ij} = \frac{1}{1 + e^{-0.1/CD_{ij}}} \quad (5.2)$$

Bu eşitliklerde, AF_{ij} anyonların çekim kuvvetini, CF_{ij} ise katyonların çekim kuvvetini ifade etmektedir. i iyon indeksini, j ise boyut indeksini temsil eder. AD_{ij} , j . boyutta i . anyonun en iyi katyondan uzaklığını, CD_{ij} ise j . boyutta i . katyonun en iyi anyondan uzaklığını ifade etmektedir. AD_{ij} ve CD_{ij} 'nin formülleri (5.3) ve (5.4) eşitliklerindeki gibidir.

$$AD_{ij} = |A_{ij} - C_{bestj}| \quad (5.3)$$

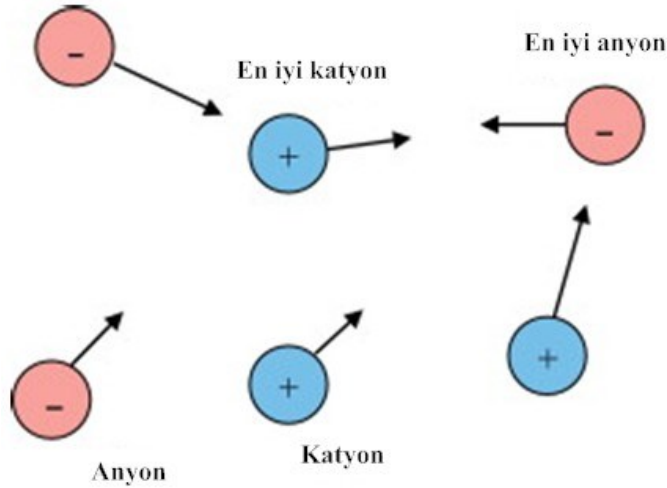
$$CD_{ij} = |C_{ij} - A_{bestj}| \quad (5.4)$$

İyonlar arasındaki çekim kuvvetleri uzaklıkla da alakalıdır. Uzaklık arttıkça çekim kuvveti azalırken uzaklık azaldıkça çekim kuvveti artar. Eşitlik (5.1) ve (5.2)'deki formüllere göre çekim kuvveti 0.5 ile 1 arasında değişen bir değer alır. Kuvvetlerin hesaplanmasından sonra anyon ve katyonların yeni konumları (5.5) ve (5.6)'daki eşitliklere göre belirlenir.

$$A_{ij} = A_{ij} + AF_{ij} \times (C_{bestj} - A_{ij}) \quad (5.5)$$

$$C_{ij} = C_{ij} + CF_{ij} \times (A_{bestj} - C_{ij}) \quad (5.6)$$

Bu eşitliklerde i iyon indeksini, j ise boyut indeksini ifade etmektedir. Her bir iyonun zıt yüklü en iyi iyon momentumu sadece güç elemanı ile belirlenmektedir. C_{best} ve A_{best} sırasıyla en iyi katyon ve anyonu ifade etmektedir. Matematiksel ifadelerden de anlaşılacağı gibi sıvı fazda rastsal bir bileşen yoktur. Şekil 5.2'de sıvı fazda iyonların hareketi gösterilmektedir.



Şekil 5.2 Anyon ve katyonların en iyi anyon ve katyona doğru yönelimleri

Zıt yüklü iyonlar birbirlerini çekme eğiliminde oldukları için algoritmada arama garanti edilir. Ancak iterasyon sayısı arttıkça iyonlar birbirlerine daha fazla yaklaşacaklar ve zıt yüklü en iyi iyonla doğru yakınsayacaklardır. Böylece, doğadaki iyonların sıvı fazdan katı faza geçişinde olduğu gibi farklılaşma/arama gitgide azalacaktır. IM

algoritmasında da iyonlar arama uzayında bir çözüme yakınsarlar, yani katı faza geçerler.

5.1.2 Katı Faz (Kuvvetlendirme)

Katı fazda, yakınsama gerçekleşmiş ve iyonlar bir optimal noktaya yakınsamıştır. Ancak, arama uzayının bilinmeyen şeklinden dolayı bu nokta yerel optimum olabilir. IM algoritmasında, yerel optimumdan kurtulmak için bir mekanizma önerilmiştir. Doğada, iyonik kristal (katı) yapıdaki anyon ve katyonlar çekimlerini artırmak için düzenlenirler. Katı fazda aynı yüklü iyonlara dış bir güç uygulandığında, üretilen itme gücü katıyı parçalar. Bu olay, yerel optimuma takılmayı çözmek için algorithmada matematiksel olarak Tablo 5.1'deki gibi modellenmiştir:

Tablo 5.1 Kristal faz sözde kodu

```

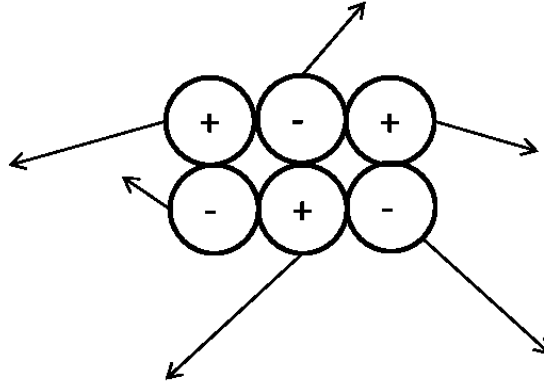
Eğer ( $C_{bestFit} \geq C_{worstFit}/2$  ve  $A_{bestFit} \geq A_{worstFit}/2$ )
{
    Eğer ( $rasgelesayı [0, 1) > 0.5$ )
         $A_i = A_i + \Phi 1 \times (C_{best} - 1)$ 
    Değilse
         $A_i = A_i + \Phi 1 \times (C_{best})$ 
    Eğer ( $rasgelesayı [0, 1) > 0.5$ )
         $C_i = C_i + \Phi 2 \times (A_{best} - 1)$ 
    Değilse
         $C_i = C_i + \Phi 2 \times (A_{best})$ 
    Eğer ( $rasgelesayı [0, 1) < 0.05$ )
         $A_i$  ve  $C_i$ 'yi rasgele oluştur.
}

```

Burada $\Phi 1$ ve $\Phi 2$ $[-1,1]$ aralığında değer alan rasgele sayıları, *rasgelesayı* $[0, 1]$ aralığında rastgele sayı türeten fonksiyonu, *CbestFit* ve *AbestFit* sırasıyla en iyi katyon ve anyonun, *CworstFit* ve *AworstFit* sırasıyla en kötü katyon ve anyonun uygunluk değerlerini ifade etmektedir.

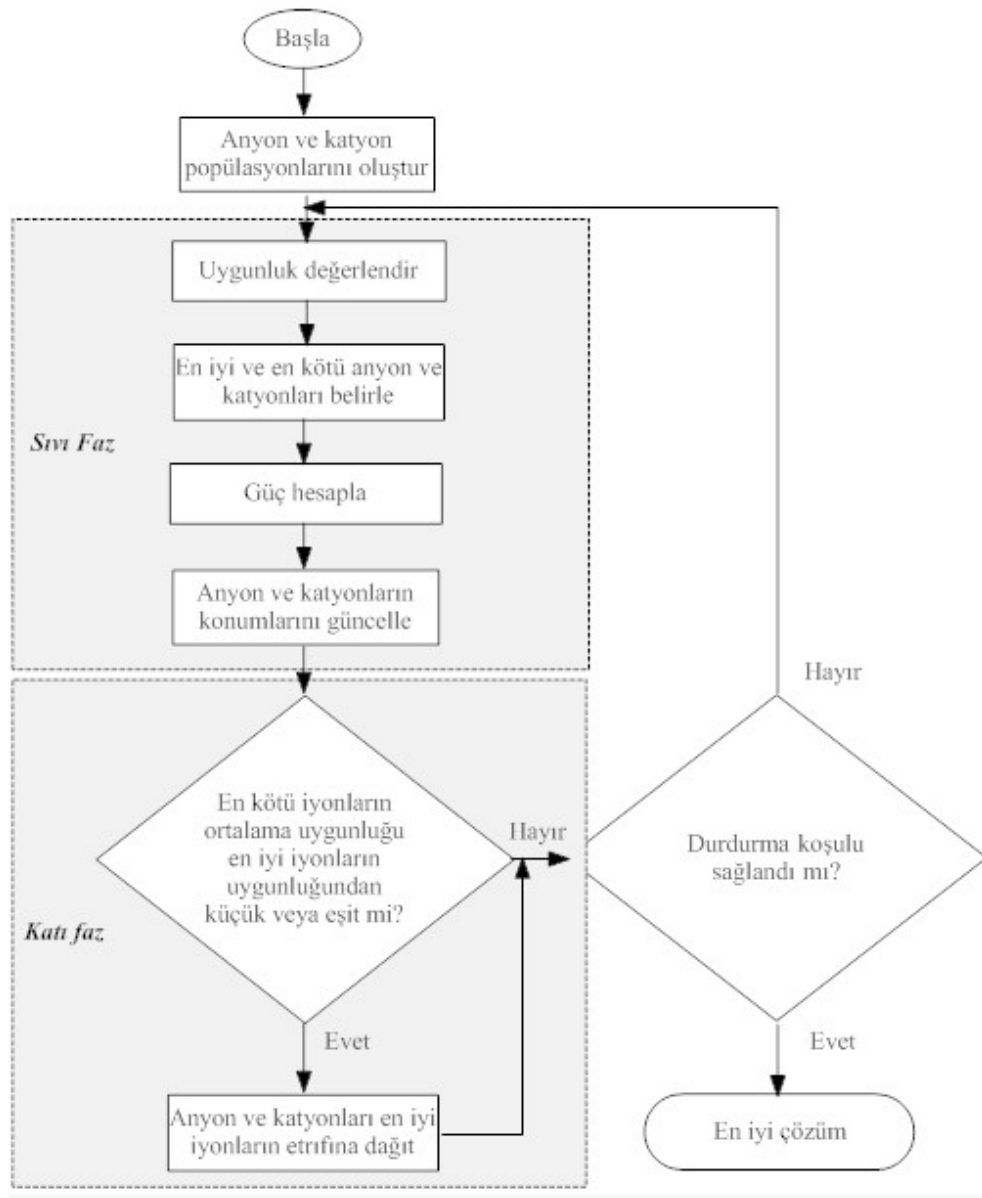
Katı faz modeline göre sıvı fazdan katı faza geçiş, en kötü iyonların uygunluk ortalamaları, en iyi iyonların uygunluklarından küçük veya eşit olduğunda mümkün olmaktadır. Bu durum sağlandığında anyon ve katyonlar sırasıyla en iyi katyon ve anyonların etrafına rastgele dağıtılmaktadır. Farklılaştırmayı artırmak için düşük bir

olasılıkla bazı anyon ve katyonlar yeniden rastgele oluşturulmaktadır. Şekil 5.3’de katı fazda iyonların dağıtımını göstermektedir.



Şekil 5.3 Yerel optimuma takılan iyonların dağıtılması

Bu fazdan sonra iyonlar tekrar sıvı faza geçmekte ve arama uzayını taramaktadır. Durdurma koşulu sağlanana kadar iyonlar katı ve sıvı faza geçmeye devam ederler. Algoritmanın çözümü yani global optimum tahmini en iyi iyondur. Algoritmanın genel akış şeması Şekil 5.4’de verilmiştir.



Şekil 5.4 İyon hareketi algoritması akış şeması

5.2 Sosyal Örümcek Algoritması

Sosyal örümcek algoritması 2015 yılında ortaya atılan yeni nesil bir metasezgisel optimizasyon algoritmasıdır [33]. Günümüzde örümcekler üzerinde çok sayıda araştırma yapılmaktadır. Örümcekler neredeyse dünyanın her yerinde yaşayan canlılardır ve genellikle tek yaşarlar. Çok nadir türleri bir grup içinde birbirleriyle etkileşim kurarak yaşarlar. Sosyal örümcek algoritması da bu örümceklerden esinlenilerek oluşturulmuş bir algoritmadır. Ayrıca örümcekler çok hassas hayvanlardır ve ağlarındaki bir değişimden, titreşimden hemen haberdar olurlar. Bu özellikleri de algorithmada kullanılmıştır.

Sosyal örümcek algoritmasında arama uzayı bir örümcek ağı olarak düşünülebilir. Ağ üzerindeki her pozisyon bir çözüme işaret etmektedir ve bütün çözümler ağ üzerinde bir yerde mevcuttur. Bu ağ ayrıca örümcekler tarafından oluşturulan titreşimleri de muhafaza etmektedir. Bütün örümcekler ağ üzerinde bir yer tutmaktadır ve o yerlerin bir amaç fonksiyonuna bağlı olarak aldığı bir değer vardır. Örümcekler ağ üzerinde hareket edebilmektedirler ve hareket edince bir titreşim oluşturmaktadırlar. Bu titreşimler sayesinde diğer örümcekler de bilgi almaktadır.

Örümcekler algoritmada kullanılan çözümlerdir ve bazı bilgileri tutmaktadırlar. Bu bilgiler:

- Ağ üzerindeki pozisyon
- Bulunulan pozisyonun değeri
- Önceki iterasyondan kalan hedef titreşim
- Hedef titreşim değerinin en son kaç iterasyon önce değiştiği
- Önceki iterasyondan sonra ne kadar hareket ettiği
- Önceki iterasyondan oluşturulan uzaklık maskesi matrisi

Titreşim sosyal örümcek algoritmasında çok önemli bir özelliktir. Titreşimin iki özelliği vardır. Bunlar titreşimin kaynağı ve titreşimin şiddetidir. Titreşimin kaynağı ağ üzerinde tanımlı herhangi bir nokta olabilir. Titreşimin şiddeti ise $[0, +\infty)$ aralığında bir değerdir. Bir örümcek hareket ettiği zaman o anki konumundan bir titreşim yayar. Bu $Pa(t)$ şeklinde ifade edilir. Buradaki a örümceğin hangi örümcek olduğunu tanımlayan indistir. t ise zamanı ifade eder. Eğer zaman o anki zaman ise sadece Pa yazmak da yeterlidir. $I(Pa, Pb, t)$ ise titreşim şiddetini ifade eder. Buradaki titreşim a örümceğinden dolayı oluşmuştur ve b örümceği bu titreşimi hisseden örümcektir. t ise zamanı ifade eder. Titreşimi oluşturan örümcek ile titreşim değerini alan örümcek aynı örümcek de olabilir. Yani $I(Pa, Pa, t)$ olarak da ifade edilen bir değer vardır ve değeri Eşitlik (5.7)'de gösterildiği gibidir.

$$I(Ps, Ps, t) = \log\left(\frac{1}{f(Ps) - C} + 1\right) \quad (5.7)$$

Buradaki $f(Ps)$, Ps pozisyonundaki amaç değerini ifade etmektedir. C ise küçük bir sabit değeri ifade etmektedir. Minimizasyon problemi için C , $f(Ps)$ 'den küçük bir pozitif sayı olmalıdır. Bu formül şu şartları sağlamaktadır:

- Bütün titreşim değerleri pozitiftir.
- Daha iyi değerdeki pozisyonlardan çıkan titreşimler, kötü pozisyonlardan çıkan titreşimlerden yüksektir.
- Global optimum noktaya yaklaşıldığı zaman titreşim miktarı çok hızlı bir şekilde artmamakta, yavaş yavaş artmaktadır.

Titreşim kuvveti uzaklıkla alakalı olarak değişiklik göstermektedir. Bu uzaklık Manhattan uzaklık formülü kullanılarak hesaplanmaktadır. İki örümcek arasındaki uzaklık Eşitlik (5.8)'deki gibidir.

$$D(Pa, Pb) = |Pa - Pb| \quad (5.8)$$

Uzaklık formülü de kullanılarak örümcekler arasında gerçekleşen titreşimler Eşitlik (5.9)'daki gibidir.

$$I(Pa, Pb, t) = I(Pa, Pa, t) \times \exp\left(-\frac{D(Pa, Pb)}{\sigma \times ra}\right) \quad (5.9)$$

Bütün örümceklerin bulundukları konumların standart sapması σ olarak ifade edilmiştir ve ra ise $[0, \infty)$ aralığında bulunan kullanıcı tanımlı bir parametredir.

Sosyal örümcek algoritması üç bölüme ayrılabilir; başlangıç, iterasyon ve sonuç. Başlangıç bölümünde amaç fonksiyonu ve çözüm uzayı tanımlanır. Örümcekler ve başlangıç pozisyonları tanımlanır. Rasgele şekilde uzaya dağıtılır. Bulundukları konumların değerleri hesaplanır. İterasyon bölümünde ise her bir iterasyonda titreşimler oluşur ve örümcekler bu titreşimlere göre yer değiştirir. Yer değiştiren örümcekler üzerinden fonksiyon değerleri hesaplanır. Bu yer değiştirmeler için diğer örümceklerden gelen titreşim değerleri birbirleriyle kıyaslanır ve en yüksek değer kaydedilir. Bütün örümcekler m adında bir matris tutarlar. Bu matris problemin boyut sayısı kadar değişken içermektedir. Ayrıca 1 ve 0'lerden oluşur. İterasyonlar geçtikçe kullanıcı tarafından belirlenen bir olasılık parametresi ile bu 1 ve 0'lar değişir. Bu m üzerinden $P_{s,i}^{fo}$, Eşitlik (5.10) ile hesaplanır.

$$P_{s,i}^{fo} = \begin{cases} P_{s,i}^{tar} & , m_{s,i} = 0 \\ P_{s,i}^r & , m_{s,i} = 1 \end{cases} \quad (5.10)$$

Buradaki r , 0 ile popülasyondaki örümcek sayısı arasında bir rasgele sayıdır.

Örümceklerin yeni konumları Eşitlik (5.11) ile hesaplanır.

$$P_s(t+1) = P_s + (P_s - P_s(t-1)) \times r + (P_{s,i}^{fo} - P_s) \times R \quad (5.11)$$

Burada R (0, 1] aralığında rasgele değerlerden oluşan bir matristir.

İterasyonlar devam ettikçe bu değerler değişir ve çözümler de artar. Sonuç bölümünde ise algoritmanın çalışma süresince elde edilen en iyi durum saklanır ve programın çıktısı olarak verilir. Algoritmanın sözde kodu Tablo 5.2'deki gibidir.

Tablo 5.2 Sosyal örümcek algoritması sözde kodu

Popülasyon büyüklüğü kadar örümcek üret
Bütün örümcekler için hedef yön (vs^{tar}) belirle
Döngü (1'den iterasyon sayısına kadar)
{
Döngü (1'den örümcek sayısına kadar)
{
Amaç fonksiyonunu hesapla
Titreşim oluştur (Eşitlik (5.7))
}
Döngü (1'den örümcek sayısına kadar)
{
Bütün örümceklerden gelen titreşimleri hesapla (Eşitlik (5.9))
Titreşimlerden en kuvvetlisini (v_{best}) seç
Eğer (en kuvvetli titreşim(v_{best}) hedef yöndeki titreşimden(v_{tar}) küçükse)
Hedef yöndeki titreşimi (v_{tar}) en iyi titreşim değeri (v_{best}) olarak belirle
Hedef titreşimin kaç iterasyondur değişmediği bilgisini (cs) güncelle
[0, 1] aralığında rasgele sayı üret
Eğer (rasgelesayı[0, 1] > Pc^{cs})
Uzaklık maskesi matrisini (m) güncelle
P_s^{fo} 'yu hesapla (Eşitlik (5.10))
Rasgele bir yere git (Eşitlik (5.11))
}
}

5.3 Stokastik Fraktal Arama Algoritması

Stokastik fraktal arama algoritması 2015 yılında ortaya atılan yeni bir algoritmadır [34]. Fraktal kelimesi Latince 'den gelmektedir ve kırılmış, çatlamış gibi anlamalara gelmektedir. Fraktallar doğadaki geometrik desenleri tanımlamak için kullanılmıştır.

Stokastik fraktal arama algoritmasında iki temel işlem vardır. Bunlar difüzyon işlemi ve güncelleme işlemidir. Difüzyon işleminde bütün parçacıklar difüzyona uğrar ve etrafında yeni parçacıklar meydana getirir. Bu işlem yerel optimum noktada takılıp kalmayı önlediği gibi global optimum noktayı bulmada da yardımcı olur. Difüzyon işlemi statik şekilde yapılmaktadır, bu difüzyon yapılan gruptaki en iyi parçacığın değerlendirilmesi diğer parçacıkların yok sayılması manasına gelir. Ayrıca çözüm uzayına etkili bir şekilde dağılmak için SFS, güncelleme kısmında bazı metotlar kullanılmaktadır.

5.3.1 Difüzyon İşlemi

Difüzyon işleminde yeni parçacıklar oluşturmak için Levy uçuşu ve Gauss olmak üzere iki istatistiksel metot yaygındır. Gauss metodu global optimum nokta bulma açısından daha başarılıdır. Bu yüzden SFS, Gauss metodu kullanılmaktadır ve hesaplamalar Eşitlik (5.12) ve Eşitlik (5.13)'e göre yapılmaktadır.

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + (\varepsilon \times BP - \varepsilon' \times P_i) \quad (5.12)$$

$$GW_2 = \text{Gaussian}(\mu_{BP}, \sigma) \quad (5.13)$$

Burada ε ve ε' $[0, 1]$ aralığında rasgele sayılardır. BP , en iyi noktayı göstermektedir. P_i ise gruptaki i 'nci sıradaki parçacığı temsil etmektedir. Standart sapma Eşitlik (5.14) ile hesaplanır.

$$\sigma = \left| \frac{\log(g)}{g} \times (P_i - BP) \right| \quad (5.14)$$

5.3.2 Güncelleme İşlemi

Grup içerisindeki bütün noktalar amaç fonksiyonunun değerine göre sıralanır ve puanlanır. Bu noktalara bir olasılık değeri atanır. Bu değer Eşitlik (5.15) ile hesaplanır.

$$Pa_i = \frac{rank(P_i)}{N} \quad (5.15)$$

Buradaki $rank(P_i)$ değeri ilgili noktanın sıralamada kaçınıcı sırada olduğudur. N ise grupta bulunan nokta sayısıdır. Bu denklem sayesinde iyi noktalarda durma olasılığı yükseltilmiş ve iyi olmayan noktaların yer değiştirme şansı artırılmıştır. Gruptaki bütün noktalar için $Pai < \varepsilon$ koşulu sağlandığı anda ilgili noktanın konumu Eşitlik (5.16)'ya göre değişecektir, aksi halde nokta yine aynı konumda kalacaktır.

$$P'_i(j) = P_r(j) - \varepsilon \times (P_t(j) - P_i(j)) \quad (5.16)$$

Burada $P'_i(j)$ yeni pozisyonu temsil etmektedir. P_r ve P_t ise grup içerisinde seçilmiş rasgele noktalar.

Bütün noktalar tekrar sıralanır ve olasılık değeri atanır. Daha sonra bu olasılık değerlerine göre Eşitlik (5.17) ve Eşitlik (5.18) uygulanır.

$$P''_i = P'_i - \varepsilon \times (P'_t - BP) \quad | \quad \varepsilon \leq 0.5 \quad (5.17)$$

$$P''_i = P'_i - \varepsilon \times (P'_t - P'_r) \quad | \quad \varepsilon > 0.5 \quad (5.18)$$

Eğer buradaki ilgili noktanın yeni konumu eski konumunda daha iyi bir durumdaysa, yeni konum eski konumun yerine geçer. Eğer eski konum yeni konumdan dahi iyiye, değişiklik gerçekleşmez.

Stokastik fraktal arama algoritmasının sözde kodu Tablo 5.3'deki gibidir.

Tablo 5.3 Stokastik fraktal arama algoritması sözde kodu

```

Popülasyon büyüklüğü kadar nokta oluştur
Döngü (1'den iterasyon sayısına kadar)
{
    Döngü (1'den popülasyon sayısına kadar bütün noktalar için)
    {
        Difüzyon işlemi
        {
            q = difüzyondan oluşan parçacık sayısı
            Döngü (1'den q sayısına kadar)
            {
                Eğer ( $\epsilon < 0.5$ )
                    Gaussian 1 formülünü uygula (Eşitlik (5.12))
                Değilse
                    Gaussian 2 formülünü uygula (Eşitlik (5.13))
            }
        }
        Güncelleme işlemi
        {
            Bütün noktaları sırala (Eşitlik (5.15))
            Döngü (1'den nokta sayısına kadar bütün noktalar için)
            {
                Döngü (her bir boyut için)
                {
                    Eğer (rasgelesayi[0, 1] >  $P_{ai}$ )
                        Yeni konum formülünü uygula (Eşitlik (5.16))
                    Değilse
                        Bir şey yapma
                }
            }
            Bütün noktaları sırala (5.15)
            Döngü (1'den bütün yeni noktalar için)
            {
                Eğer (rasgelesayi [0, 1] >  $P_{ai}$ )
                    İkinci yeni konum formüllerini uygula (Eşitlik (5.17)
                    (5.18))
                Değilse
                    Bir şey yapma
            }
        }
    }
}

```

6. BÖLÜM

YENİ NESİL METASEZGİSELLERLE YSA EĞİTİMİ

İyon hareketi algoritması, sosyal örümcek algoritması ve stokastik fraktal arama algoritması yapay sinir ağlarının eğitimi için uygun hale getirilmiş, Java programlama dili ile bilgisayar ortamında kodlanmış ve test edilmiştir. Algoritmaların yapay sinir ağları eğitimindeki performanslarını değerlendirmek için UCI makine öğrenme deposundan (<http://archive.ics.uci.edu/ml>) alınan 7 veri kümesi kullanılmıştır. Veri kümelerinin temel özellikleri Tablo 6.1’de gösterilmektedir. Her bir veri kümesi Karşılaştırmalar hem yeni nesil algoritmaların kendi aralarında hem de literatürde mevcut diğer bazı klasik YSA eğitim ve metasezgisel algoritmalarla yapılmıştır.

Tablo 6.1 Veri kümelerinin temel özellikleri

Veri kümesi	Örnek sayısı	Nitelik sayısı	Sınıf sayısı
Cancer	699	9	2
Card	690	51	2
Diabetes	768	2	2
Gene	3175	120	3
Glass	214	9	6
Soybean	683	82	19
Thyroid	7200	21	3

6.1 Veri Kümeleri

Cancer veri kümesi: Göğüs kanseri ile ilgili verileri içerir. Kanser tipinin iyi huylu mu yoksa kötü huylu mu olduğunu bulmaya çalışır. 9 adet girdi ve 2 adet çıktıya sahiptir. 699 adet de örnek mevcuttur. Bu verilerden 525 tanesi eğitim için, geri kalan 174 tanesi de test için kullanılmıştır.

Card veri kümesi: Müşterilerin kredi kartı başvurularını onaylayıp onaylamama durumlarını içeren bir veri kümesidir. 51 adet girdisi ve 2 adet çıktısı vardır. 690 adet

örneğe sahip olmakla beraber, bunların 518 tanesi eğitim verisi, 192 tanesi ise test verisi olarak kullanılmıştır.

Diabetes veri kümesi: Diyabet hastalığının pozitif veya negatif çıkmasını tutan bir veri kümesidir. 8 adet girdisi ve 2 adet çıktısı vardır. 576 tanesi eğitim kümesi ve 192 tanesi test kümesi olmak üzere toplam 768 tane veriye sahiptir.

Gene veri kümesi: Gen dizilimlerini analiz eden bir çalışmayı tutan veri kümesidir. 120 adet girdiye ve 3 adet çıktıya sahiptir. Bu veri kümesinde 3715 adet örnek yer almaktadır. Bunlardan 2382 tanesi eğitim, 793 tanesi ise test verileridir.

Glass veri kümesi: Camların türünü ifade eden verilerden oluşmaktadır. 9 adet girdi, 6 adet çıktı ve 214 adet örneğe sahiptir. Bu verilerden 161 tanesi eğitim, geri kalan 53 tanesi ise test verisi olarak kullanılmıştır.

Soybean veri kümesi: Soyları farklı türlere ayıran özellikleri tutan bir veri kümesidir. 82 adet girdi, 19 adet çıktı ve 693 adet örneğe sahiptir. Bu verilerden 513 tanesi eğitim için 170 tanesi de test için kullanılmıştır.

Thyroid veri kümesi: Tiroit hastalığını sınıflandıran bir veri kümesidir. 21 adet girdisi ve 3 adet çıktısı mevcuttur. Toplam 7200 adet örnek içermektedir ve bunlardan 5400 tanesi eğitim, geri kalan 1800 tanesi de test kümesi olarak kullanılmıştır.

6.2 Çalışmada Kullanılan Yeni Metasezgiseller ve Karşılaştırmada Kullanılan Algoritmalar

Bu çalışmada kullanılan ve yapay sinir ağları eğitimine uyarlanan yeni metasezgisel algoritmalar şunlardır:

- İyon hareketi algoritması
- Sosyal örümcek algoritması
- Stokastik fraktal arama algoritması

Yapay sinir ağları eğitimi için daha önceden kullanılan ve yeni nesil metasezgiseller ile karşılaştırılan algoritmalar ise şunlardır [35, 36]:

- Yapay arı kolonisi algoritması
- Parçacık sürü optimizasyonu algoritması
- Diferansiyel gelişim algoritması (DE)
- Genetik algoritmalar
- Harmoni arama algoritması
- Av arama algoritması
- Geri yayılım algoritması
- Levenberg Marquardt algoritması

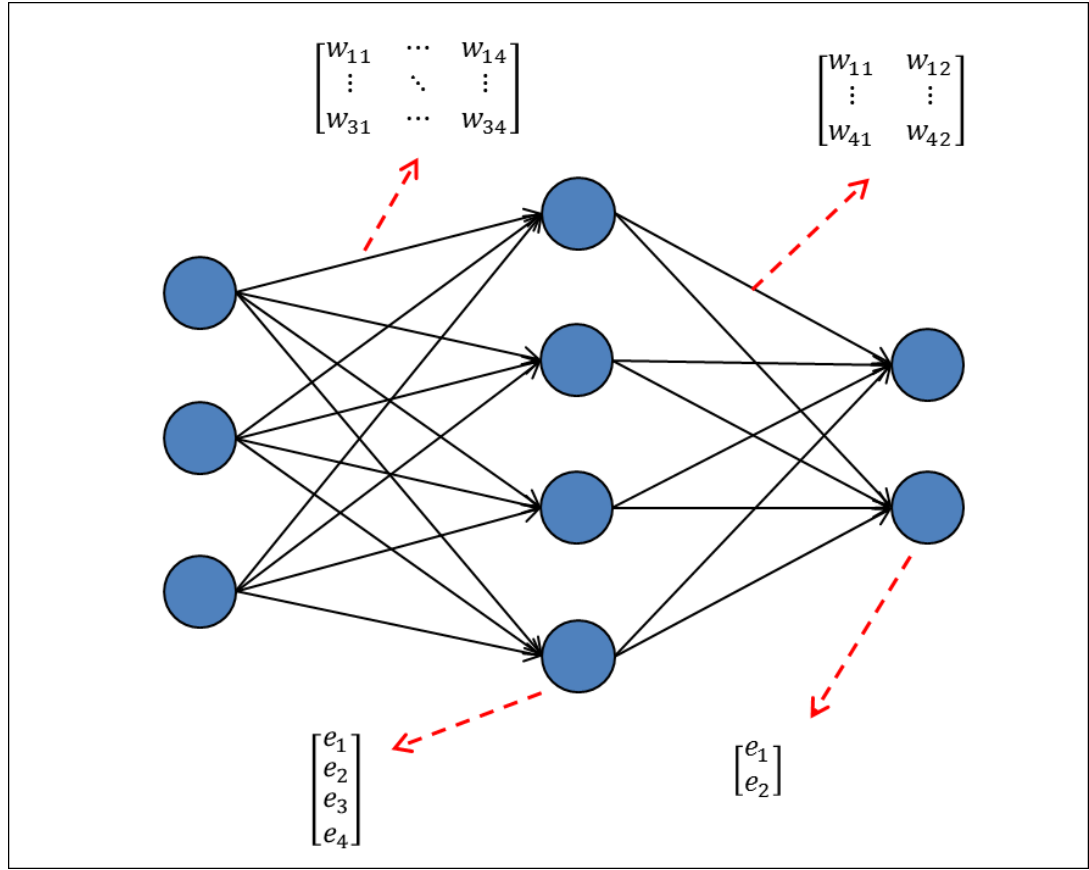
6.3 Dizi Gösterimi

Yapay sinir ağları eğitimi, ilgili YSA'nın nöronları arasındaki ağırlıklarının değiştirilmesiyle gerçekleştirilmektedir. Bu ağırlık değerleri değiştikçe bilgilerin ağ içerisindeki dağılımları, dolayısıyla çıktı değerleri değişmektedir. Ağırlıkların hatayı en küçükleyecek şekilde değiştirilmesiyle sınıflandırma işlemini gerçekleştirmektedir.

Tez çalışmasında ele alınan yeni nesil metasezgisel algoritmalarda popülasyondaki her bir dizi, YSA ağırlıklarını ifade etmektedir. Ağırlık değerleri matrisler aracılığıyla tutulmaktadır ve matris boyutları Tablo 6.2'de verilmiştir. Şekil 6.1'de 3 girdi 2 çıktılı bir problem için gizli katmanında 4 nöron içeren YSA yapısında ağırlık matrisleri gösterilmektedir. Şekilden de görüldüğü gibi girdi-gizli katman ağırlıklarının tutulduğu matris 3×4 boyutunda, gizli-çıkı ağırlıklarının tutulduğu matris 4×2 boyutunda, gizli katman eşik ağırlıklarının tutulduğu matris 6×1 boyutunda ve çıktı eşik ağırlıklarının tutulduğu matris 2×1 boyutundadır.

Tablo 6.2 YSA eğitimi için tutulan ağırlık matrisleri

Katmanlar	Boyut
Girdi katmanı ile gizli katman arası	Girdi sayısı $\times$ gizli katmandaki nöron sayısı
Gizli katman ile çıktı katmanı arası	Gizli katmandaki nöron sayısı $\times$ çıktı sayısı
Gizli katman eşik değerleri	Gizli katman nöron sayısı
Çıktı katmanı eşik değerleri	Çıktı sayısı



Şekil 6.1 YSA eğitimi için tutulan ağırlık değerleri matrisleri

6.4 YSA Tasarımı ve Veri Kümelerinin Eğitim-Test Kümelerine Ayrılması

Algoritmaların sağlıklı bir şekilde karşılaştırılması için yapay sinir ağlarının tasarımının aynı olması gerekmektedir. Bunun için yeni algoritmalar için tasarlanan yapay sinir ağları daha önceki algoritmalarda kullanılan şekilde tasarlanmıştır. Bütün veri kümeleri için girdi ve çıktı arasında 1 gizli katman vardır ve bu gizli katman 6 adet nöron içermektedir. Aktivasyon fonksiyonu olarak ise sigmoid aktivasyon fonksiyonu kullanılmıştır.

Karşılaştırmada ele alınan algoritmalarda veri kümesinin eğitim ve test kümesine parçalanmasında “basit bölümlleme” kullanılmış ve veri kümelerinin %75’lik kısmı eğitim verisi olarak, geri kalan %25’lik kısmı ise test verisi olarak kullanılmıştır. Karşılaştırmaların daha doğru ve adil olması adına IM, SS ve SFS algoritmaları için de tüm veri kümeleri aynı şekilde eğitim ve test kümelerine parçalanmıştır.

6.5 Etkinlik Ölçütleri

Algoritmalar veriyi ne kadar oranda hatalı sınıflandırıldığında göre değerlendirilmektedir. Hatalı sınıflandırma oranı Eşitlik (6.1) kullanılarak hesaplanmaktadır. Ayrıca algoritmaların çalışma süreleri de algoritmaların etkinliklerini değerlendirmede kullanılmaktadır.

$$\text{Hatalı sınıflandırma oranı} = 100 \times \frac{\text{Hatalı sınıflandırılmış veri sayısı}}{\text{Veri sayısı}} \quad (6.1)$$

6.6 Deney Tasarımı

Ele alınan yeni nesil metasezgisel algoritmalar farklı parametrelere sahiptir ve bu parametrelerin aldığı değerlere göre çözüm etkinlikleri değişmektedir. YSA eğitiminde bu parametrelerden etkin olanlarını ve parametrelerin en uygun değerlerini belirlemek amacıyla bu bölümde bir deney tasarımı gerçekleştirilmiştir. Çalışmada ele alınan metasezgisel algoritmalar çok fazla parametreye sahip olmadığı için, tam faktöriyel deney tasarımları Minitab 16 istatistik paket programı vasıtasıyla düzenlenmiştir. Deneyler on kez tekrarlı şekilde yapılmış ve her bir parametre için 2 düzeyli bir tasarım gerçekleştirilmiştir.

Tam faktöriyel tasarımında hatalı sınıflandırma oranı üzerinde faktörlerin etkileri analiz edilirken, ana etkiler ve etkileşimler dikkate alınmıştır. Çalışılacak çok sayıda tasarım faktörü varsa çok kullanışlı olan ana etki grafikleri, düzey ortalamalarını karşılaştırmaya ve etkinlik ölçütünü en çok etkileyen faktörün ayırt edilmesine yardımcı olmaktadır. Ana etki grafiğinde elde edilen doğru x-eksenine paralel ise, faktörün ele alınan her düzeyi, ölçütü aynı yönde etkilemektedir ve ana etki yoktur. Eğer elde edilen doğru belirli bir eğime sahipse, bu durumda bir ana etki mevcuttur ve ele alınan faktörün farklı düzeyleri, etkinlik ölçütünü farklı şekilde etkilemektedir. Doğrudaki eğim arttıkça etkinin önemi de artar. Ana etki grafiklerinde, doğruların eğimleri karşılaştırılarak, faktörün göreceli önemi belirlenebilir.

Ancak ana etki grafikleri faktörlerin etkinlikleri hakkında net bir sonuca varmak için yeterli değildir. Çünkü etkileşim eğrileri, ana etkiyi doğrulayabilecekleri gibi ortadan da kaldırılabirler. Bu bakımdan etkileşim grafiklerinin de incelenmesi gerekir. Bir etkileşim grafiği bir faktörün ayarlamalarını diğer bir faktöre değiştirmekteki etkiyi

gösterir. Etkileşim grafikleri ana etkiyi doğrulayabileceğinden veya ortadan kaldırılabileceğinden dolayı, etkileşimleri değerlendirmek çok önemlidir. Eğer etkileşim grafiklerinde ele alınan faktörlerin doğruları paralele yakınsa, bu durumda bu faktörler birbirinden bağımsızdır ve çok az veya hiç etkileşim yoktur şeklinde yorumlanır. Kesişen doğrular, faktörler arasındaki güçlü etkileşimleri ifade ederler.

Deney tasarımı sonucu ana etkiler, ikili etkileşim grafikleri ve küp grafiklerden faydalanılarak hatalı sınıflandırma oranı üzerinde etkisi olan faktörler ve bu faktörlerin etkin düzeyleri belirlenmiştir. Çalışmada ele alınan üç metasezgisel algoritma, her bir veri kümesi için faktörlerin belirlenen düzeylerinde çalıştırılmış ve etkinlik ölçütleri hesaplanmıştır. Deneysel tasarımda analizlerin her bir veri kümesi için ayrı ayrı ele alınmasındaki neden, veri kümesindeki örnek sayısının değişmesi ile faktörlerin etkinliğinin değişebileceğidir.

6.6.1 İyon Hareketi Algoritması Deney Tasarımı

İyon hareketi algoritmasında popülasyon büyüklüğü ve iterasyon sayısı olmak üzere iki parametre mevcuttur. Bu durum algoritmanın en büyük avantajlarından birisidir. Deneyde test edilen parametreler ve parametre düzeyleri Tablo 6.3’de gösterildiği gibidir.

Tablo 6.3 İyon hareketi algoritması deney tasarımı parametreleri

Parametre	Değerler
Popülasyon büyüklüğü	50 – 100
İterasyon sayısı	500 – 1000

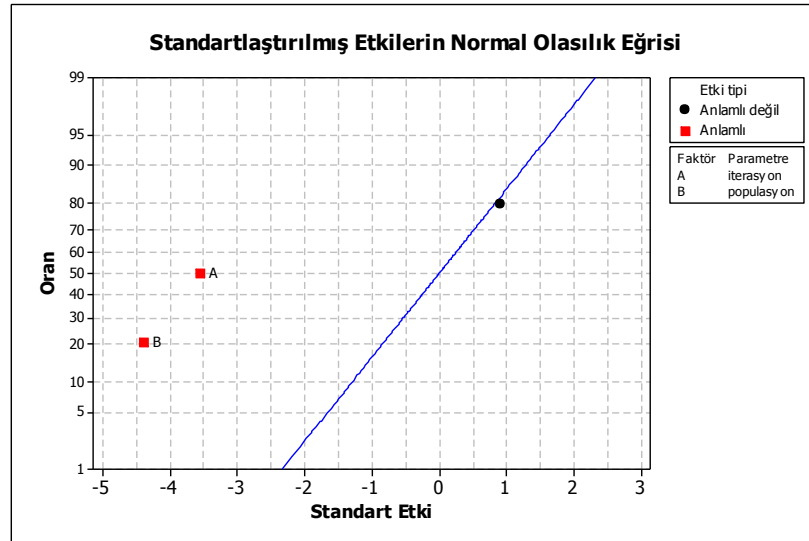
6.6.1.1 Gene Veri Kümesi

IM algoritması Gene veri kümesi üzerinde deney tasarımı sonuçları Tablo 6.4’de gösterilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında iterasyon sayısı ve popülasyon büyüklüğü parametreleri tek başına etkilidir ancak etkileşimleri söz konusu değildir.

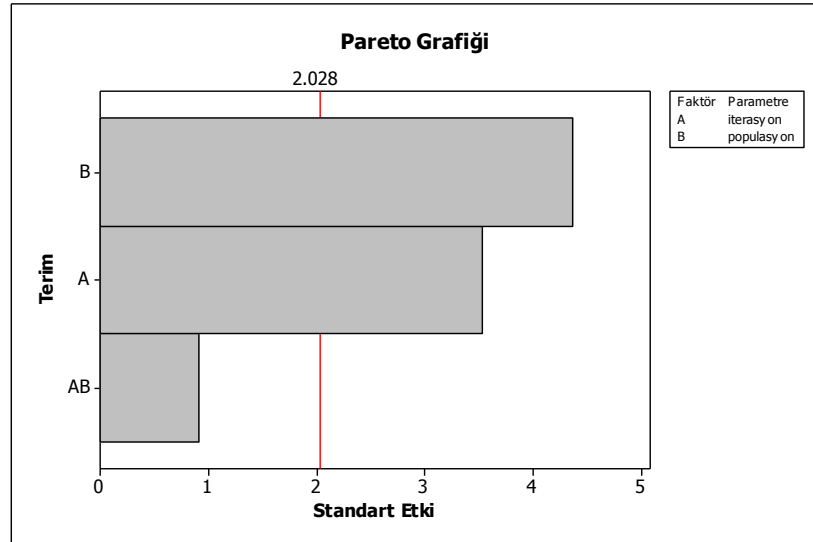
Tablo 6.4 IM algoritması Gene veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		24.3	0.7627	31.86	0	
iterasyon	-5.397	-2.699	0.7627	-3.54	0.001	
populasyon	-6.671	-3.335	0.7627	-4.37	0	
iterasyon*populasyon	1.4	0.7	0.7627	0.92	0.365	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	736.31	736.306	368.15	15.82	0
İkili_Etkileşimler	1	19.59	19.593	19.59	0.84	0.365
Artık_Hata	36	837.69	837.686	23.27		
Toplam	39	1593.58				

Şekil 6.2 normal olasılık eğrisinde, kırmızı ile gösterilen populasyon büyüklüğü ve iterasyon sayısı etkin parametreler olarak görülmektedir. Aynı şekilde Şekil 6.3’de verilen pareto grafiğinde de bu iki parametre kırmızı çizgi ile gösterilen standart etkinin üzerinde çıkmıştır ve etkilidir. Bu sonuçlar Tablo 6.4’de elde edilen sonucu desteklemektedir.



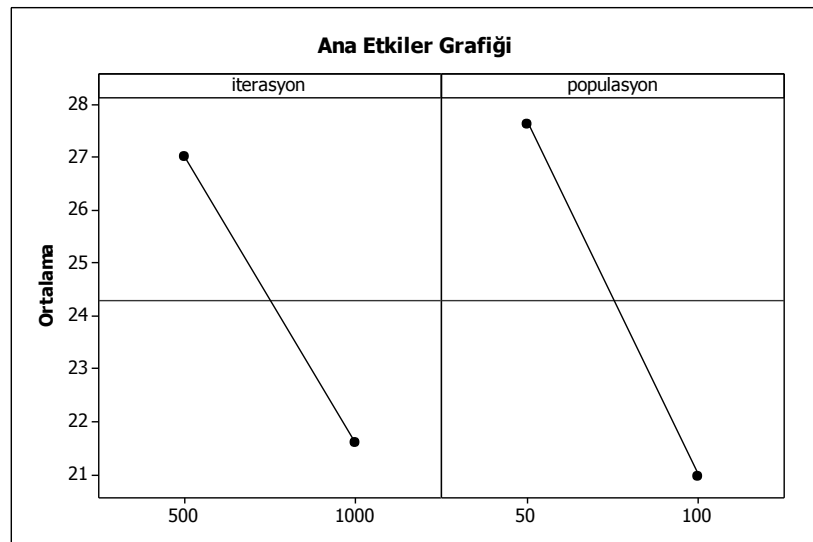
Şekil 6.2 IM algoritması Gene veri kümesi normal olasılık eğrisi



Şekil 6.3 IM algoritması Gene veri kümesi pareto grafiği

Hatalı sınıflandırma oranı ölçütü üzerinde etkinliğe sahip olan parametrelerin en uygun düzeylerini belirlemek için ana etki ve etkileşim grafiklerini incelemek gerekir. Şekil 6.4 IM algoritması Gene veri kümesinde ana etkiler grafiğini göstermektedir. Ele alınan uygunluk ölçütü bir minimizasyon ölçütü olduğu için ana etkiler grafiğine göre parametre düzeyleri şu şekilde belirlenmiştir.

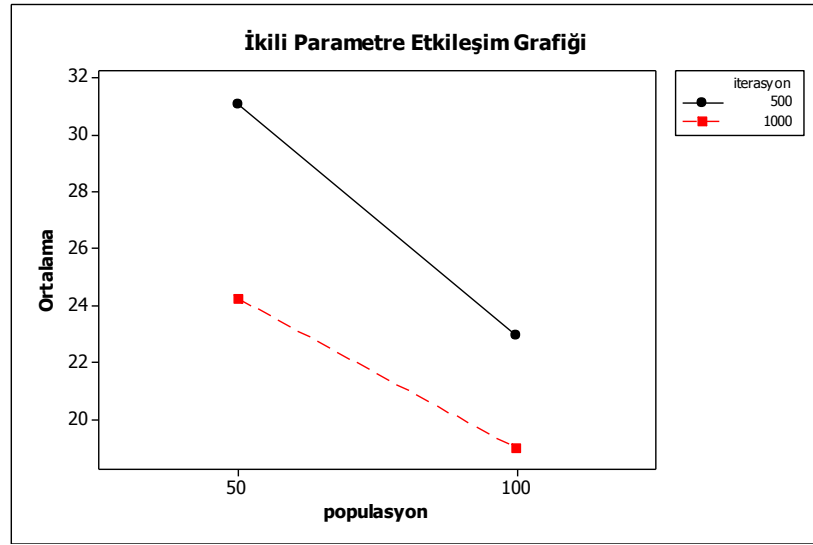
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.4 IM algoritması Gene veri kümesi tekli parametre etkileri

Şekil 6.5 IM algoritması Gene veri kümesinde ikili etkileşim grafiğini göstermektedir. Grafiğe göre parametreler arasında bir etkileşim yoktur ve parametre düzeyleri ana etkiler grafiği ile örtüşür şekilde aşağıdaki gibi belirlenmiştir.

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.5 IM algoritması Gene veri kümesi ikili parametre etkileri

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. İterasyon sayısı ve popülasyon büyüklüğü parametrelerinin her ikisi de Gene veri kümesinde etkili çıktığı için yukarıda belirtilen düzeyler kullanılmalıdır.

6.6.1.2 Cancer Veri Kümesi

Cancer veri kümesi için deney tasarımı sonuçları Tablo 6.5'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında parametreler tek başlarına ve ikili etkileşimli hallerine göre anlamlı değildir. Yani popülasyon büyüklüğünün 50 veya 100 olması ve iterasyon sayısının 500 veya 1000 olması uygunluk ölçütünü önemli bir şekilde etkilememektedir.

Tablo 6.5 IM algoritması Cancer veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		0.53161	0.0392	13.56	0	
İterasyon	0.02874	0.01437	0.0392	0.37	0.716	
Popülasyon	-0.02874	-0.01437	0.0392	-0.37	0.716	
iterasyon*populasyon	-0.02874	-0.01437	0.0392	-0.37	0.716	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	0.01651	0.01651	0.008257	0.13	0.875
İkili_Etkileşimler	1	0.00826	0.00826	0.008257	0.13	0.716
Artık_Hata	36	2.21297	2.21297	0.061471		
Toplam	39	2.23775				

IM algoritması Cancer veri kümesi için deney tasarımından elde edilen Normal olasılık eğrisi, pareto grafiği, ana etkiler ve etkileşim grafikleri ekler bölümünde verilmiştir.

6.6.1.3 Card Veri Kümesi

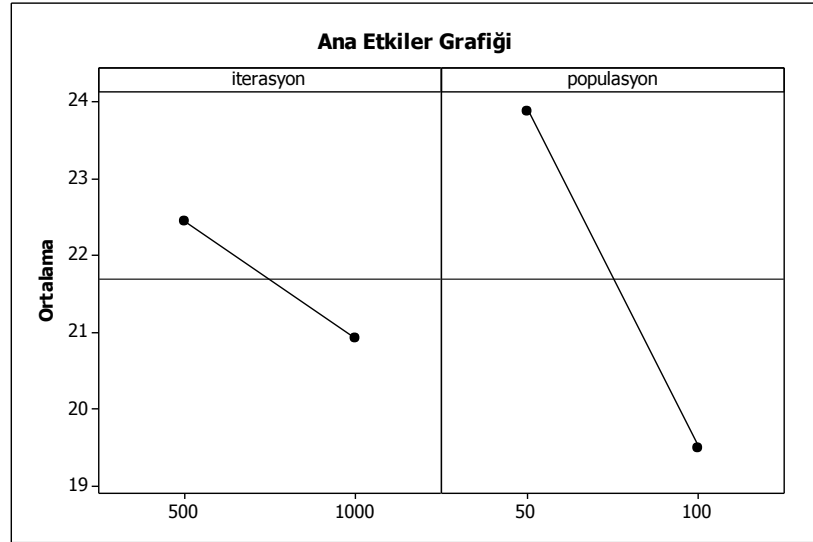
IM algoritması Card veri kümesi üzerinde deney tasarımı sonuçları Tablo 6.6'daki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında, p değeri=0 ile (<0.05) popülasyon büyüklüğü etkili bir parametredir.

Tablo 6.6 IM algoritması Card veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		21.686	0.4805	45.14	0	
iterasyon	-1.512	-0.756	0.4805	-1.57	0.124	
populasyon	-4.36	-2.18	0.4805	-4.54	0	
iterasyon*populasyon	-0.756	-0.378	0.4805	-0.79	0.437	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	212.987	212.987	106.493	11.53	0
İkili_Etkileşimler	1	5.713	5.713	5.713	0.62	0.437
Artık_Hata	36	332.409	332.409	9.234		
Toplam	39	551.109				

Şekil 6.6’da gösterilen ana etki grafikleri incelendiğinde hatalı sınıflandırma oranı uygunluk ölçütü bakımından parametrelerin ideal değerleri şöyle belirlenmiştir:

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.6 IM algoritması Card veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği ve ikili etkileşim grafikleri ekler bölümünde verilmiştir. Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri:

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Card veri kümesinde tek etkili parametre popülasyon büyüklüğü olduğu için popülasyon büyüklüğü 100 düzeyinde kullanılmalıdır. İterasyon sayısı için diğer düzey de kullanılabilir.

6.6.1.4 Diabetes Veri Kümesi

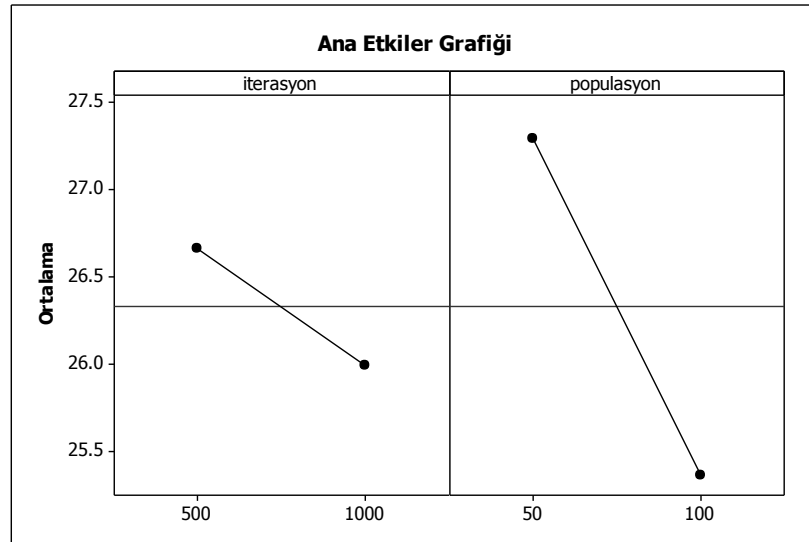
Deney tasarımı sonuçları Tablo 6.7’deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında popülasyon büyüklüğü α anlamlılık düzeyinden küçük çıktığı için tek başına etkilidir.

Tablo 6.7 IM algoritması Diabetes veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		26.3281	0.2045	128.74	0	
iterasyon	-0.6771	-0.3385	0.2045	-1.66	0.107	
populasyon	-1.9271	-0.9635	0.2045	-4.71	0	
iterasyon*populasyon	0.0521	0.026	0.2045	0.13	0.899	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	41.721	41.7209	20.8605	12.47	0
İkili_Etkileşimler	1	0.027	0.0271	0.0271	0.02	0.899
Artık_Hata	36	60.221	60.2214	1.6728		
Toplam	39	101.969				

Şekil 6.7 ana etkiler grafiğini göstermektedir. Bu grafiğe göre parametre düzeyleri aşağıdaki gibi belirlenmelidir:

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.7 IM algoritması Diabetes veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği ve ikili etkileşim grafikleri ekler bölümünde verilmiştir.

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri:

- İterasyon sayısı: 1000

- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Tek etkili parametre popülasyon büyüklüğü olduğu için popülasyon büyüklüğü olarak 100 düzeyi kullanılmalıdır. İterasyon sayısı parametresi için iki düzeyden herhangi biri kullanılabilir.

6.6.1.5 Glass Veri Kümesi

Deney tasarımı sonuçları Tablo 6.8'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında tasarım sonuçlarına göre popülasyon büyüklüğü tek başına etkilidir.

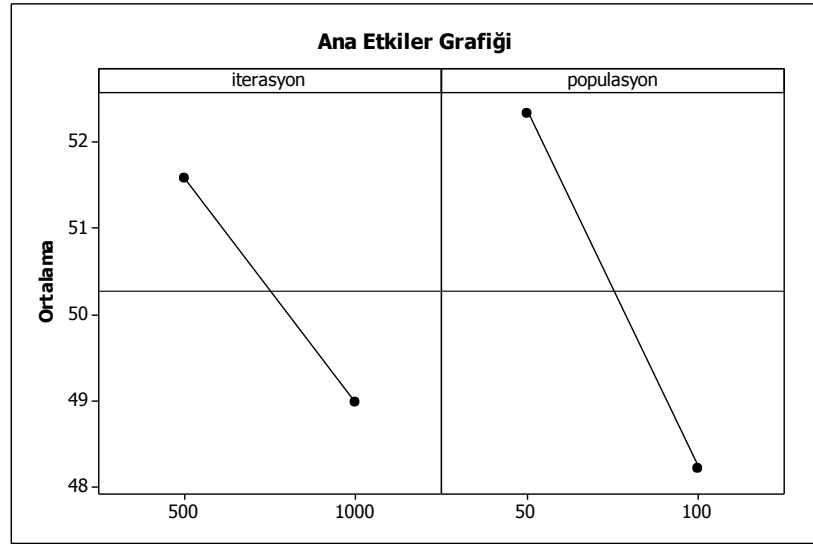
Tablo 6.8 IM algoritması Glass veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		50.274	0.9019	55.74	0	
iterasyon	-2.603	-1.301	0.9019	-1.44	0.158	
populasyon	-4.11	-2.055	0.9019	-2.28	0.029	
iterasyon*populasyon	0.137	0.068	0.9019	0.08	0.94	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	236.63	236.63	118.315	3.64	0.036
İkili_Etkileşimler	1	0.19	0.19	0.188	0.01	0.94
Artık_Hata	36	1171.33	1171.33	32.537		
Toplam	39	1408.14				

Şekil 6.8 ana etkiler grafiğini göstermektedir. Ana etkiler grafiğine göre parametrelerin ideal değerleri

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmektedir.



Şekil 6.8 IM algoritması Glass veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği ve ikili etkileşim grafikleri ekler bölümünde verilmiştir.

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Veri kümesi için tek etkili parametre popülasyon büyüklüğü çıktığı için popülasyon büyüklüğü parametresi için 100 düzeyi kullanılmalıdır.

6.6.1.6 Soybean Veri Kümesi

IM algoritması Soybean veri kümesi üzerinde deney tasarımı sonuçları Tablo 6.9’de gösterilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında popülasyon büyüklüğü tek başına etkilidir.

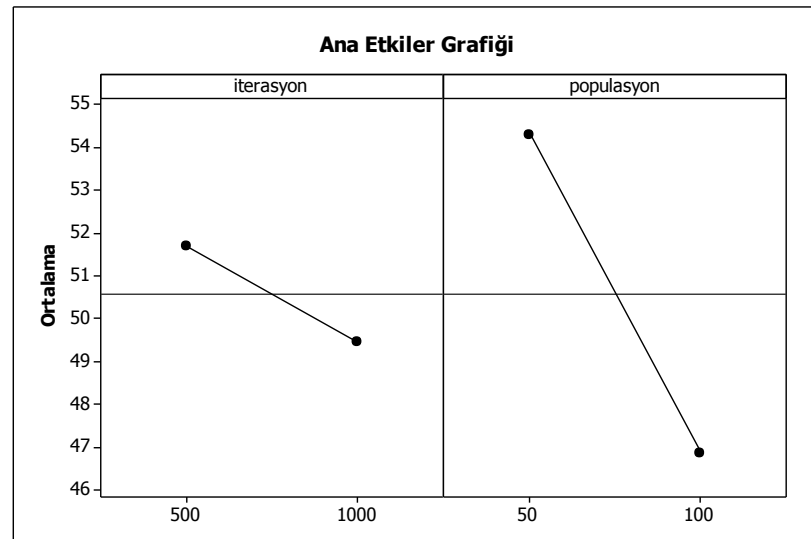
Tablo 6.9 IM algoritması Soybean veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		50.58	0.865	58.47	0	
İterasyon	-2.232	-1.116	0.865	-1.29	0.205	
Popülasyon	-7.411	-3.705	0.865	-4.28	0	
iterasyon*populasyon	0.982	0.491	0.865	0.57	0.574	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	599.01	599.01	299.506	10.01	0
İkili_Etkileşimler	1	9.65	9.65	9.646	0.32	0.574
Artık_Hata	36	1077.49	1077.49	29.93		
Toplam	39	1686.14				

Şekil 6.9’de gösterilen ana etkiler grafiği incelendiğinde parametrelerin farklı düzeylerinin hatalı sınıflandırma oranı üzerinde etkili olduğu görülmektedir. Ana etkiler grafiğine göre parametrelerin ideal değerleri

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir.



Şekil 6.9 IM algoritması Soybean veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği ve ikili etkileşim grafikleri ekler bölümünde verilmiştir.

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Soybean veri kümesi için de tek etkili parametre popülasyon büyüklüğü çıkmıştır ve parametre düzeyi olarak 100 kullanılmalıdır.

6.6.1.7 Thyroid Veri Kümesi

Deney tasarımı sonuçları Tablo 6.10’da verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında etkili parametre yoktur.

Tablo 6.10 IM algoritması Thyroid veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		3.9345	0.1449	27.15	0	
İterasyon	-0.2815	-0.1408	0.1449	-0.97	0.338	
Popülasyon	-0.3661	-0.1831	0.1449	-1.26	0.215	
iterasyon*populasyon	-0.2261	-0.113	0.1449	-0.78	0.44	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	2	2.1328	2.1328	1.0664	1.27	0.293
İkili_Etkileşimler	1	0.5111	0.5111	0.5111	0.61	0.44
Artık_Hata	36	30.2382	30.2382	0.84		
Toplam	39	32.8821				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ekler bölümünde verilmiştir. Thyroid veri kümesi için etkili parametre çıkmadığı için parametreler iki düzeyinden biri ile kullanılabilir.

IM algoritması için çalışmada ele alınan 7 veri kümesinden Cancer ve Thyroid veri kümelerinde etkili parametre çıkmamıştır. Diğer veri kümelerinde Gene veri kümesi hariç hepsinde sadece popülasyon büyüklüğü parametresi etkilidir ve tasarım sonucunda düzey olarak 100 değeri belirlenmiştir. Gene veri kümesinde popülasyon büyüklüğünün yanı sıra iterasyon sayısı da etkili parametre olarak çıkmıştır. İlgili veri kümesinde tasarım sonucu iterasyon sayısı 1000 ve popülasyon büyüklüğü 100 olarak belirlenmiştir. Dolayısıyla IM algoritması bütün veri kümeleri için bundan sonraki

deneysel çalışmalar ve karşılaştırmalarda parametre düzeyleri olarak aşağıdaki değerler kullanılmıştır.

- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

6.6.2 Sosyal Örümcek Algoritması Deney Tasarımı

Sosyal örümcek algoritmasında popülasyon büyüklüğü, iterasyon sayısı, ra, pm ve pc olmak üzere beş parametre mevcuttur. Deneyde test edilen parametreler ve parametre düzeyleri Tablo 6.11’da gösterildiği gibidir.

Tablo 6.11 Sosyal örümcek algoritması deney tasarımı parametreleri

Parametreler	Değerler
Popülasyon büyüklüğü	50 – 100
İterasyon sayısı	500 – 1000
Ra	0.1 – 10
Pc	0.2 – 0.7
Pm	0.1 – 0.6

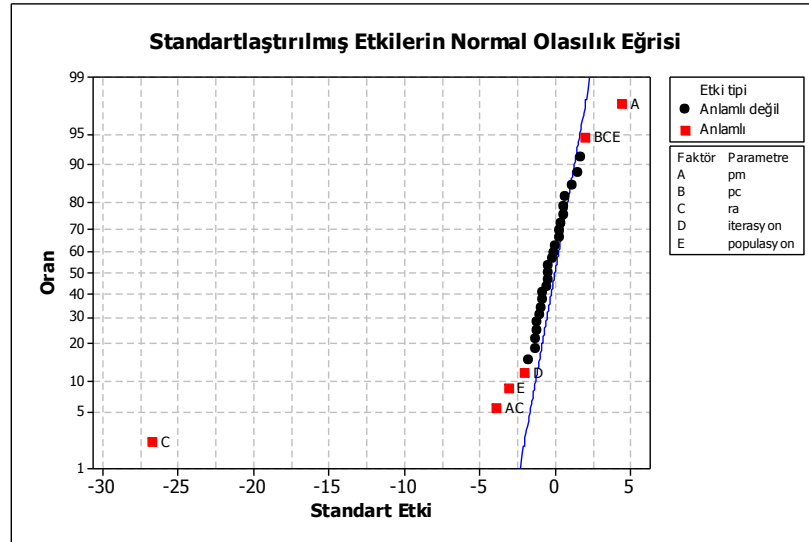
6.6.2.1 Cancer Veri Kümesi

Deney tasarımı sonuçları Tablo 6.12’deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında pm, ra, iterasyon sayısı ve popülasyon büyüklüğü parametreleri hatalı sınıflandırma oranı üzerinde etkilidir. İkili etkileşimler açısından pm ile ra arasında; üçlü etkileşimler açısından ise pc, ra ve popülasyon büyüklüğü arasında bir etkileşim olduğu görülmektedir.

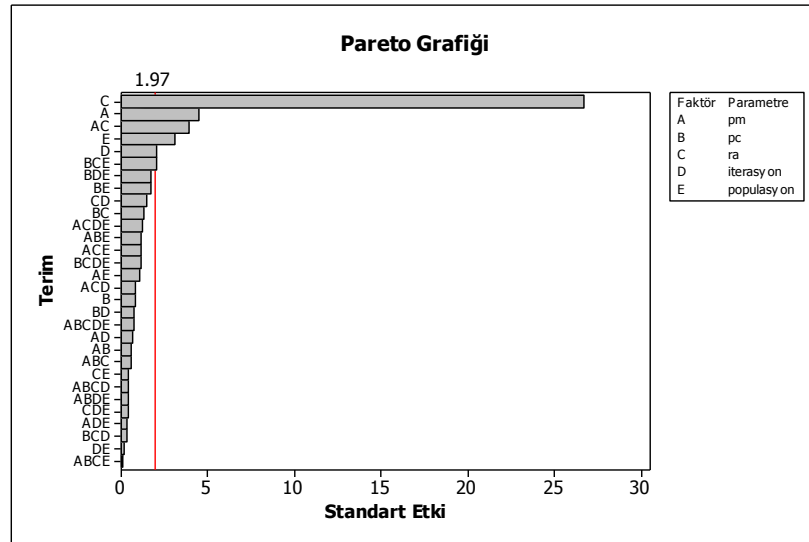
Tablo 6.12 SS algoritması Cancer veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		7.184	0.2102	34.18	0	
pm	1.868	0.934	0.2102	4.44	0	
pc	-0.338	-0.169	0.2102	-0.8	0.422	
ra	-11.221	-5.611	0.2102	-26.7	0	
iterasyon	-0.848	-0.424	0.2102	-2.02	0.045	
populasyon	-1.293	-0.647	0.2102	-3.08	0.002	
pm*pc	0.251	0.126	0.2102	0.6	0.55	
pm*ra	-1.638	-0.819	0.2102	-3.9	0	
pm*iterasyon	0.273	0.136	0.2102	0.65	0.517	
pm*populasyon	-0.431	-0.216	0.2102	-1.03	0.306	
pc*ra	-0.539	-0.269	0.2102	-1.28	0.201	
pc*iterasyon	-0.323	-0.162	0.2102	-0.77	0.442	
pc*populasyon	-0.711	-0.356	0.2102	-1.69	0.092	
ra*iterasyon	0.632	0.316	0.2102	1.5	0.134	
ra*populasyon	0.187	0.093	0.2102	0.44	0.657	
iterasyon*populasyon	-0.057	-0.029	0.2102	-0.14	0.891	
pm*pc*ra	-0.223	-0.111	0.2102	-0.53	0.597	
pm*pc*iterasyon	-0.022	-0.011	0.2102	-0.05	0.959	
pm*pc*populasyon	-0.496	-0.248	0.2102	-1.18	0.239	
pm*ra*iterasyon	-0.359	-0.18	0.2102	-0.85	0.393	
pm*ra*populasyon	0.489	0.244	0.2102	1.16	0.246	
pm*iterasyon*populasyon	0.144	0.072	0.2102	0.34	0.733	
pc*ra*iterasyon	0.136	0.068	0.2102	0.32	0.746	
pc*ra*populasyon	0.841	0.42	0.2102	2	0.046	
pc*iterasyon*populasyon	0.726	0.363	0.2102	1.73	0.085	
ra*iterasyon*populasyon	-0.158	-0.079	0.2102	-0.38	0.707	
pm*pc*ra*iterasyon	-0.18	-0.09	0.2102	-0.43	0.669	
pm*pc*ra*populasyon	0.036	0.018	0.2102	0.09	0.932	
pm*pc*iterasyon*populasyon	-0.18	-0.09	0.2102	-0.43	0.669	
pm*ra*iterasyon*populasyon	-0.517	-0.259	0.2102	-1.23	0.219	
pc*ra*iterasyon*populasyon	-0.481	-0.241	0.2102	-1.15	0.253	
pm*pc*ra*iterasyon*populasyon	0.295	0.147	0.2102	0.7	0.484	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	10552.8	10552.8	2110.56	149.3	0
İkili_Etkileşimler	10	347.6	347.6	34.76	2.46	0.008
Üçlü_Etkileşimler	10	156.8	156.8	15.68	1.11	0.354
Dörtlü_Etkileşimler	5	45.2	45.2	9.04	0.64	0.67
Beşli_Etkileşimler	1	6.9	6.9	6.94	0.49	0.484
Artık_Hata	288	4070.3	4070.3	14.13		
Toplam	319	15179.7				

Şekil 6.10 ve Şekil 6.11’de verilen normal olasılık eğrisi ve pareto grafiği incelendiğinde de etkili olan parametreler yukarıda bulunan sonuçla aynı çıkmaktadır.



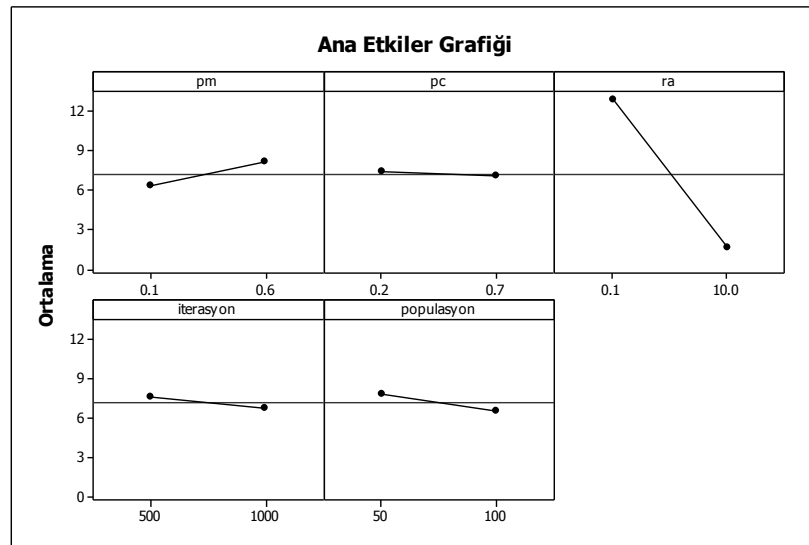
Şekil 6.10 SS algoritması Cancer veri kümesi normal olasılık eğrisi



Şekil 6.11 SS algoritması Cancer veri kümesi pareto grafiği

Parametrelerin etkin olduğu düzeyleri belirlemek için hazırlanan ve Şekil 6.12’de gösterilen ana etkiler grafiği incelendiğinde parametre düzeyleri aşağıdaki gibi belirlenmiştir.

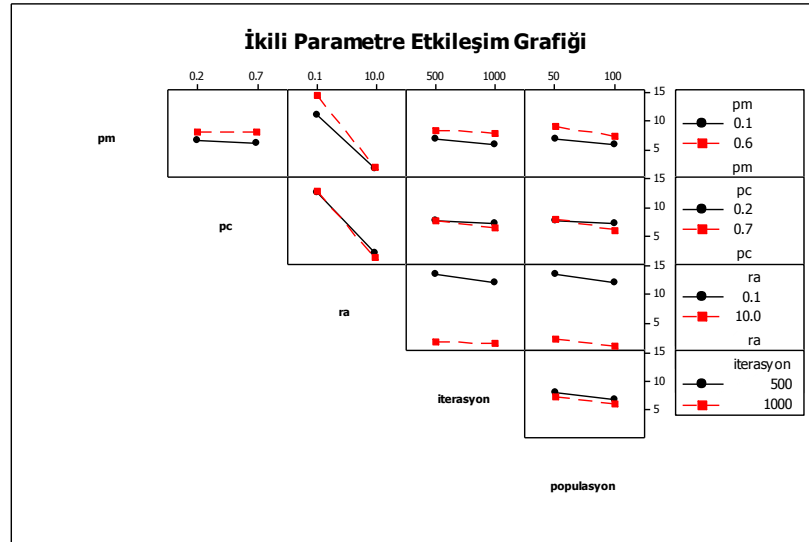
- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.12 SS algoritması Cancer veri kümesi tekli parametre etkileri

Şekil 6.13 SS algoritması Cancer veri kümesi üzerinde etkileşim grafiğini göstermektedir. Grafik incelendiğinde, pm ile ra parametreleri ve pc ile popülasyon büyüklüğü parametreleri arasında ikili etkileşim olduğu görülmektedir. Etkileşim grafikleri dikkate alınarak parametre düzeyleri aşağıdaki gibi belirlenmiştir.

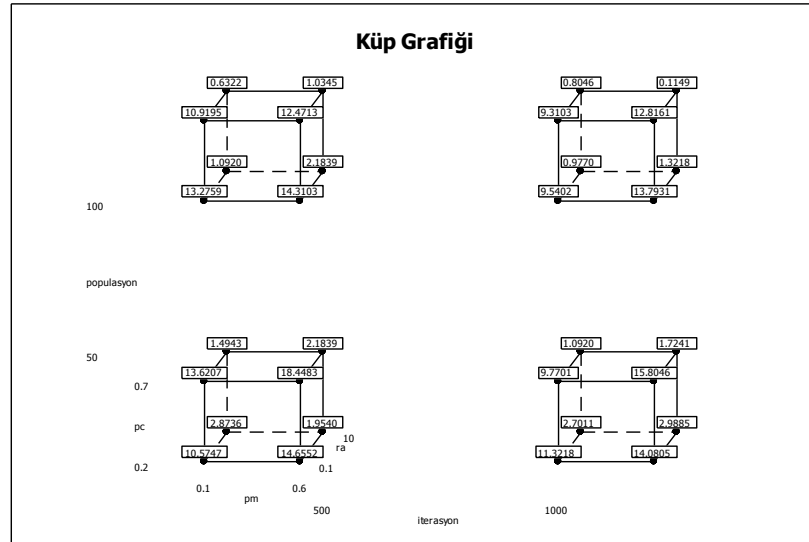
- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.13 SS algoritması Cancer veri kümesi ikili parametre etkileri

Çoklu etkileşimleri incelemek için küp grafikten yararlanılır. Küp grafik okunurken öncelikle parametrelerin tek başlarına ve ikili etkileşimleri incelenir ve en iyi değerleri belirlenir. Bu değerler üzerinden küp grafik üzerinde bakılacak yerler seçilir. Sonrasında ise en düşük değere bakılır ve bu değeri sağlayan üçlü etkileşim parametre değerleri saptanır. Şekil 6.14’de gösterilen küp grafik üzerinden parametrelerin üçlü etkileşimleri incelendiğinde parametre düzeyleri olarak yine daha önceki elde edilen düzeyler belirlenmektedir.

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.14 SS algoritması Cancer veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda Cancer veri kümesi için en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Gerek ana etkiler gerekse etkileşimler açısından tüm parametrelerin hatalı sınıflandırma oranı ölçütü üzerinde etkisi vardır. Dolayısıyla yukarıda verilen parametre düzeylerinin hepsi dikkate alınmalıdır.

6.6.2.2 Card Veri Kümesi

Deney tasarımı sonuçları Tablo 6.13'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında pm ve ra tek başlarına etkilidir. pm ve ra ayrıca ikili etkileşimle çözüm üzerinde etkilidir. Bunların dışında pm, pc ve popülasyon büyüklüğü arasında bir üçlü etkileşim söz konusudur.

Tablo 6.13 SS algoritması Card veri kümesi deney tasarımı sonuçları

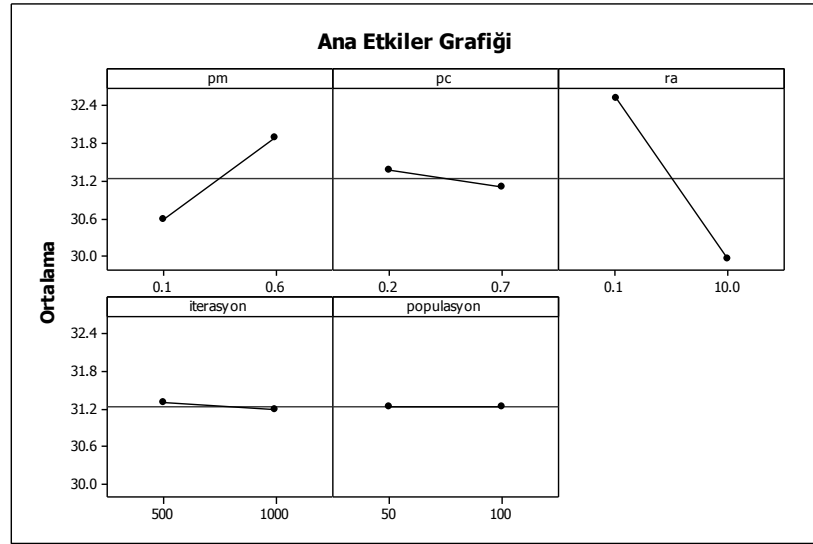
Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		31.23	0.1858	168.09	0	
pm	1.297	0.649	0.1858	3.49	0.001	
pc	-0.287	-0.144	0.1858	-0.77	0.44	
ra	-2.569	-1.285	0.1858	-6.91	0	
iterasyon	-0.113	-0.056	0.1858	-0.3	0.762	
populasyon	0.018	0.009	0.1858	0.05	0.961	
pm*pc	-0.142	-0.071	0.1858	-0.38	0.703	
pm*ra	0.774	0.387	0.1858	2.08	0.038	
pm*iterasyon	0.498	0.249	0.1858	1.34	0.181	
pm*populasyon	0.309	0.154	0.1858	0.83	0.407	
pc*ra	0.367	0.184	0.1858	0.99	0.324	
pc*iterasyon	0.498	0.249	0.1858	1.34	0.181	
pc*populasyon	0.105	0.053	0.1858	0.28	0.777	
ra*iterasyon	0.178	0.089	0.1858	0.48	0.632	
ra*populasyon	0.382	0.191	0.1858	1.03	0.305	
iterasyon*populasyon	0.222	0.111	0.1858	0.6	0.551	
pm*pc*ra	-0.127	-0.064	0.1858	-0.34	0.732	
pm*pc*iterasyon	0.062	0.031	0.1858	0.17	0.868	
pm*pc*populasyon	1.239	0.62	0.1858	3.33	0.001	
pm*ra*iterasyon	0.062	0.031	0.1858	0.17	0.868	
pm*ra*populasyon	-0.171	-0.085	0.1858	-0.46	0.646	
pm*iterasyon*populasyon	0.687	0.343	0.1858	1.85	0.066	
pc*ra*iterasyon	-0.302	-0.151	0.1858	-0.81	0.418	
pc*ra*populasyon	0.018	0.009	0.1858	0.05	0.961	
pc*iterasyon*populasyon	0.294	0.147	0.1858	0.79	0.429	
ra*iterasyon*populasyon	-0.578	-0.289	0.1858	-1.55	0.121	
pm*pc*ra*iterasyon	0.193	0.096	0.1858	0.52	0.605	
pm*pc*ra*populasyon	-0.534	-0.267	0.1858	-1.44	0.152	
pm*pc*iterasyon*populasyon	0.178	0.089	0.1858	0.48	0.632	
pm*ra*iterasyon*populasyon	-0.491	-0.245	0.1858	-1.32	0.188	
pc*ra*iterasyon*populasyon	-0.287	-0.144	0.1858	-0.77	0.44	
pm*pc*ra*iterasyon*populasyon	0.178	0.089	0.1858	0.48	0.632	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	670.26	670.26	134.052	12.14	0
İkili_Etkileşimler	10	126.59	126.59	12.659	1.15	0.328
Üçlü_Etkileşimler	10	205.74	205.74	20.574	1.86	0.05
Dörtlü_Etkileşimler	5	54.17	54.17	10.835	0.98	0.43
Beşli_Etkileşimler	1	2.54	2.54	2.536	0.23	0.632
Artık_Hata	288	3181.21	3181.21	11.046		
Toplam	319	4240.51				

SS algoritması Card veri kümesi için oluşturulan normal olasılık eğrisi ve pareto grafiği ekler bölümünde verilmiştir. Etkili parametrelerin en iyi düzeylerine karar vermek için Şekil 6.15’de gösterilen ana etkiler grafiği incelendiğinde ideal değerler

- Pm: 0.1
- Pc: 0.7

- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 50 veya 100

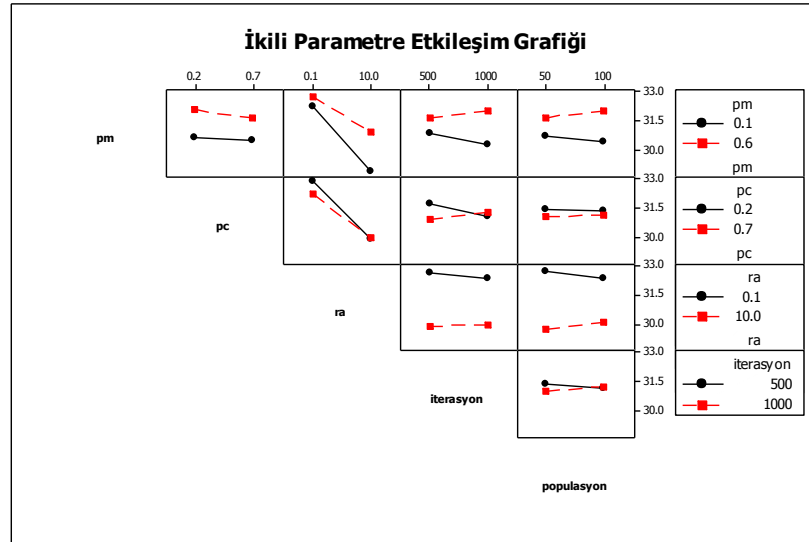
olarak belirlenmektedir.



Şekil 6.15 SS algoritması Card veri kümesi tekli parametre etkileri

Şekil 6.16'da verilen etkileşim grafiği üzerinden parametreler arasındaki ikili etkileşimler değerlendirildiğinde ise parametre düzeyleri aşağıdaki şekilde belirlenmektedir.

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 50 veya 100

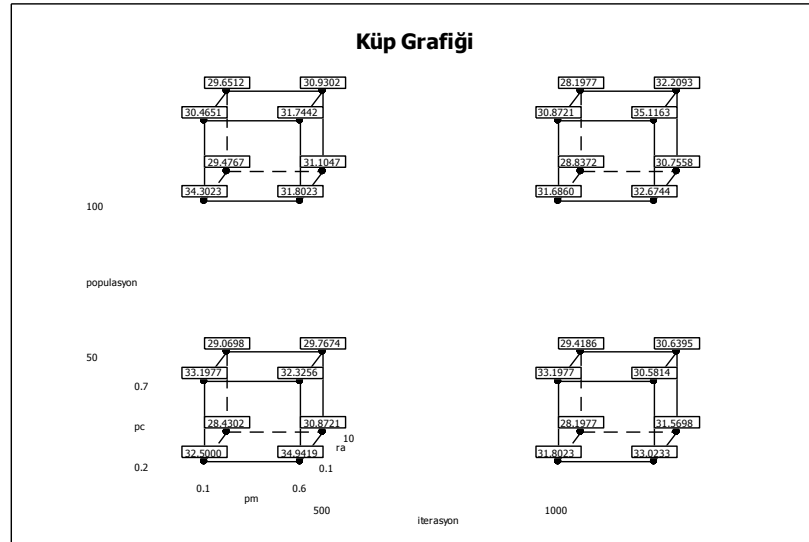


Şekil 6.16 SS algoritması Card veri kümesi ikili parametre etkileri

Şekil 6.17’de gösterilen küp grafik üzerinden parametrelerin üçlü etkileşimleri incelendiğinde ideal düzeyler

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmektedir.



Şekil 6.17 SS algoritması Card veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda Card veri kümesinde en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Algoritma parametrelerinden iterasyon sayısı hariç diğer tüm parametreler gerek ana etkiler gerekse etkileşimler bakımından etkilidir. Dolayısıyla iterasyon sayısı hariç diğer tüm parametrelerin düzeyleri yukarıdaki değerlerinde kullanılmalıdır. İterasyon sayısı ise iki düzeyden herhangi birinde kullanılabilir.

6.6.2.3 Diabetes Veri Kümesi

Deney tasarımı sonuçları Tablo 6.14'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında pm ve ra tek başlarına etkilidir. Pm ile ra ve iterasyon sayısı ile popülasyon büyüklüğü ikili etkileşimle çözüm üzerinde etkilidir. Bunların dışında pm, pc, ra ve iterasyon sayısı parametreleri arasında ise bir dördümlü etkileşim söz konusudur.

Tablo 6.14 SS algoritması Diabetes veri kümesi deney tasarımı sonuçları

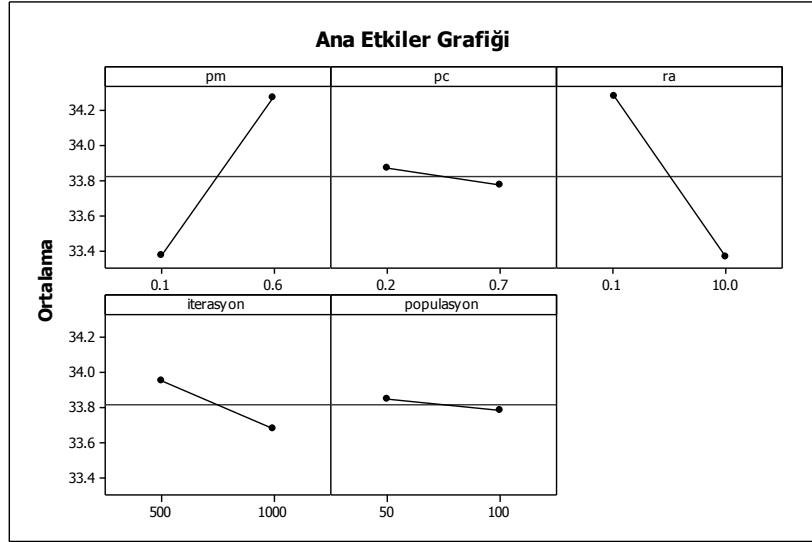
Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		33.8184	0.0825	409.9	0	
pm	0.8984	0.4492	0.0825	5.44	0	
pc	-0.0911	-0.0456	0.0825	-0.55	0.581	
ra	-0.918	-0.459	0.0825	-5.56	0	
iterasyon	-0.2734	-0.1367	0.0825	-1.66	0.099	
populasyon	-0.0586	-0.0293	0.0825	-0.36	0.723	
pm*pc	0.2799	0.14	0.0825	1.7	0.091	
pm*ra	0.5208	0.2604	0.0825	3.16	0.002	
pm*iterasyon	0.1237	0.0618	0.0825	0.75	0.454	
pm*populasyon	0.1693	0.0846	0.0825	1.03	0.306	
pc*ra	0.0391	0.0195	0.0825	0.24	0.813	
pc*iterasyon	0.0456	0.0228	0.0825	0.28	0.783	
pc*populasyon	0.0911	0.0456	0.0825	0.55	0.581	
ra*iterasyon	-0.0521	-0.026	0.0825	-0.32	0.753	
ra*populasyon	-0.1367	-0.0684	0.0825	-0.83	0.408	
iterasyon*populasyon	0.3646	0.1823	0.0825	2.21	0.028	
pm*pc*ra	0.1497	0.0749	0.0825	0.91	0.365	
pm*pc*iterasyon	0.1302	0.0651	0.0825	0.79	0.431	
pm*pc*populasyon	-0.0456	-0.0228	0.0825	-0.28	0.783	
pm*ra*iterasyon	-0.0716	-0.0358	0.0825	-0.43	0.665	
pm*ra*populasyon	-0.013	-0.0065	0.0825	-0.08	0.937	
pm*iterasyon*populasyon	-0.1888	-0.0944	0.0825	-1.14	0.253	
pc*ra*iterasyon	-0.293	-0.1465	0.0825	-1.78	0.077	
pc*ra*populasyon	0.0781	0.0391	0.0825	0.47	0.636	
pc*iterasyon*populasyon	0.0716	0.0358	0.0825	0.43	0.665	
ra*iterasyon*populasyon	-0.2083	-0.1042	0.0825	-1.26	0.208	
pm*pc*ra*iterasyon	0.4167	0.2083	0.0825	2.53	0.012	
pm*pc*ra*populasyon	0.0456	0.0228	0.0825	0.28	0.783	
pm*pc*iterasyon*populasyon	-0.1953	-0.0977	0.0825	-1.18	0.238	
pm*ra*iterasyon*populasyon	0.0716	0.0358	0.0825	0.43	0.665	
pc*ra*iterasyon*populasyon	0.2148	0.1074	0.0825	1.3	0.194	
pm*pc*ra*iterasyon*populasyon	-0.0521	-0.026	0.0825	-0.32	0.753	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	138.909	138.909	27.7818	12.75	0
İkili_Etkileşimler	10	44.786	44.786	4.4786	2.06	0.028
Üçlü_Etkileşimler	10	17.829	17.829	1.7829	0.82	0.611
Dörtlü_Etkileşimler	5	21.21	21.21	4.2419	1.95	0.087
Beşli_Etkileşimler	1	0.217	0.217	0.217	0.1	0.753
Artık_Hata	288	627.333	627.333	2.1782		
Toplam	319	850.284				

SS algoritması Diabetes veri kümesi için oluşturulan normal olasılık eğrisi ve pareto grafiği ekler bölümünde verilmiştir. Şekil 6.18 ana etkiler grafiğini göstermektedir. Ana etkiler grafiğine göre parametrelerin ideal düzeyleri:

- Pm: 0.1
- Pc: 0.7

- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

şeklinde belirlenmektedir.

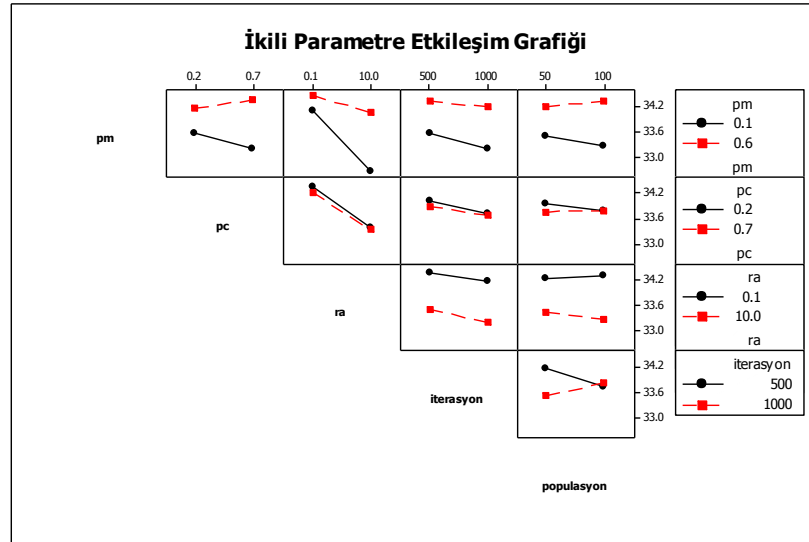


Şekil 6.18 SS algoritması Diabetes veri kümesi tekli parametre etkileri

Şekil 6.19’da verilen etkileşim grafiğinden ikili etkileşimler incelendiğinde parametrelerin düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

şeklinde belirlenmiştir.

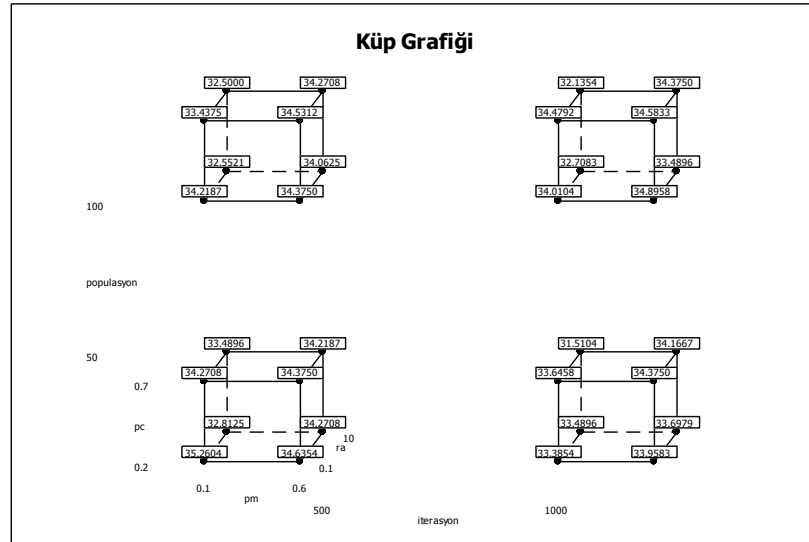


Şekil 6.19 SS algoritması Diabetes veri kümesi ikili parametre etkileşimleri

Şekil 6.20’de gösterilen küp grafik üzerinden parametrelerin dörtlü etkileşimleriyle incelendiğinde ideal değerler

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmektedir.



Şekil 6.20 SS algoritması Diabetes veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Diabetes veri kümesinde parametrelerin hepsi gerek ana etkiler gerekse etkileşimler bakımından etkilidir. Dolayısıyla tüm parametreler yukarıda verilen değerlerinde kullanılmalıdır.

6.6.2.4 Gene Veri Kümesi

Deney tasarımı sonuçları Tablo 6.15'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında ra tek başına etkilidir. Pm ile ra ve ra ile popülasyon büyüklüğü ikili etkileşimle çözüm üzerinde etkilidir.

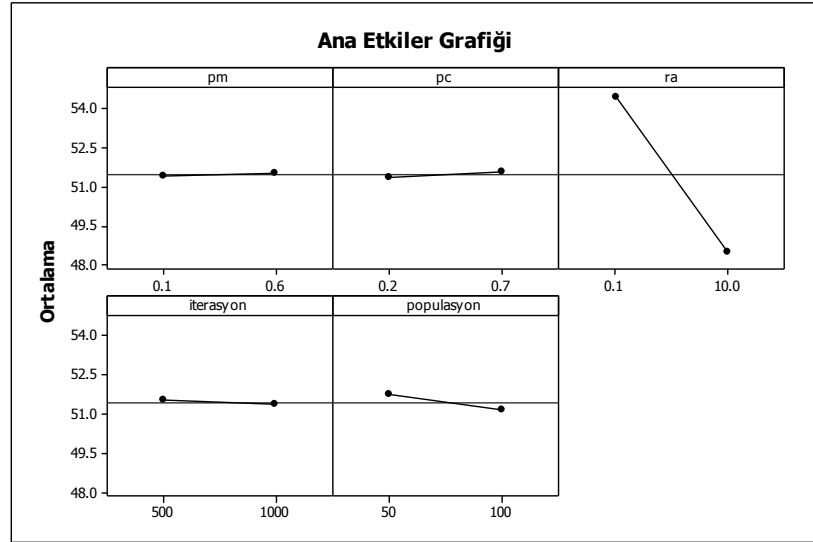
Tablo 6.15 SS algoritması Gene veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		51.439	0.2662	193.21	0	
pm	0.133	0.067	0.2662	0.25	0.803	
pc	0.225	0.112	0.2662	0.42	0.673	
ra	-5.953	-2.976	0.2662	-11.18	0	
iterasyon	-0.152	-0.076	0.2662	-0.29	0.775	
populasyon	-0.582	-0.291	0.2662	-1.09	0.275	
pm*pc	0.541	0.271	0.2662	1.02	0.31	
pm*ra	1.924	0.962	0.2662	3.61	0	
pm*iterasyon	-0.113	-0.056	0.2662	-0.21	0.833	
pm*populasyon	0.34	0.17	0.2662	0.64	0.524	
pc*ra	-0.545	-0.272	0.2662	-1.02	0.307	
pc*iterasyon	0.335	0.167	0.2662	0.63	0.53	
pc*populasyon	-0.628	-0.314	0.2662	-1.18	0.239	
ra*iterasyon	-0.612	-0.306	0.2662	-1.15	0.251	
ra*populasyon	1.114	0.557	0.2662	2.09	0.037	
iterasyon*populasyon	0.209	0.104	0.2662	0.39	0.695	
pm*pc*ra	-0.527	-0.264	0.2662	-0.99	0.323	
pm*pc*iterasyon	-0.19	-0.095	0.2662	-0.36	0.722	
pm*pc*populasyon	0.505	0.253	0.2662	0.95	0.344	
pm*ra*iterasyon	0.256	0.128	0.2662	0.48	0.631	
pm*ra*populasyon	-0.382	-0.191	0.2662	-0.72	0.473	
pm*iterasyon*populasyon	0.731	0.365	0.2662	1.37	0.171	
pc*ra*iterasyon	0.099	0.049	0.2662	0.19	0.853	
pc*ra*populasyon	0.314	0.157	0.2662	0.59	0.555	
pc*iterasyon*populasyon	0.75	0.375	0.2662	1.41	0.16	
ra*iterasyon*populasyon	-0.507	-0.253	0.2662	-0.95	0.342	
pm*pc*ra*iterasyon	-0.001	0	0.2662	0	0.999	
pm*pc*ra*populasyon	-0.378	-0.189	0.2662	-0.71	0.479	
pm*pc*iterasyon*populasyon	0.896	0.448	0.2662	1.68	0.093	
pm*ra*iterasyon*populasyon	-0.001	0	0.2662	0	0.999	
pc*ra*iterasyon*populasyon	-0.713	-0.357	0.2662	-1.34	0.181	
pm*pc*ra*iterasyon*populasyon	-0.721	-0.361	0.2662	-1.35	0.177	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	2869.4	2869.38	573.88	25.3	0
İkili_Etkileşimler	10	526.8	526.78	52.68	2.32	0.012
Üçlü_Etkileşimler	10	179.4	179.36	17.94	0.79	0.638
Dörtlü_Etkileşimler	5	116.3	116.35	23.27	1.03	0.402
Beşli_Etkileşimler	1	41.6	41.61	41.61	1.83	0.177
Artık_Hata	288	6532.1	6532.08	22.68		
Toplam	319	10265.6				

SS algoritması Gene veri kümesi için oluşturulan normal olasılık eğrisi ve pareto grafiği ekler bölümünde verilmiştir. Şekil 6.21’de gösterilen ana etkiler grafiği üzerinden parametreler incelendiğinde ideal düzeyler aşağıdaki şekilde belirlenmektedir.

- Pm: 0.1
- Pc: 0.2

- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

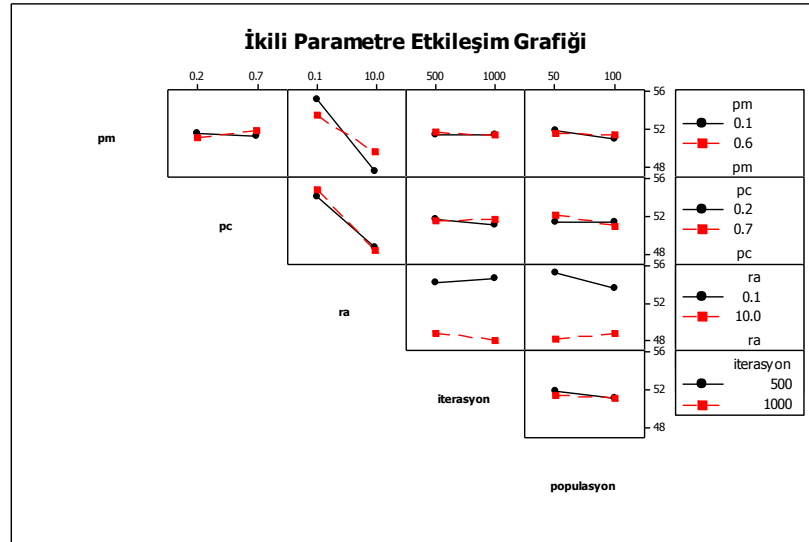


Şekil 6.21 SS algoritması Gene veri kümesi tekli parametre etkileri

Şekil 6.22’de verilen etkileşim grafiği üzerinden parametrelerin ikili etkileşimleri incelendiğinde parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

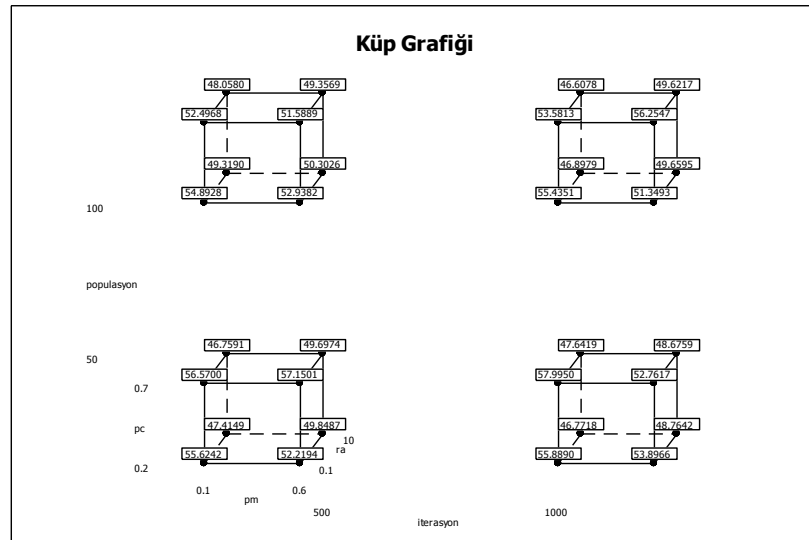
şeklinde belirlenmektedir.



Şekil 6.22 SS algoritması Gene veri kümesi ikili parametre etkileşimleri

Şekil 6.23’de gösterilen parametrelerin küp grafiği incelendiğinde parametrelerin ideal düzeyleri aşağıdaki gibi belirlenmiştir.

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.23 SS algoritması Gene veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Yukarıda verilen parametrelerden pm, ra ve popülasyon büyüklüğü parametreleri ana etkiler veya etkileşimler açısından etkili çıkmıştır. Dolayısıyla bu parametreler yukarıda verilen düzeylerinde kullanılmalıdır.

6.6.2.5 Glass Veri Kümesi

Deney tasarımı sonuçları Tablo 6.16'daki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında pm ve ra tek başlarına etkilidir. Pm ve ra ayrıca ikili etkileşimle çözüm üzerinde etkilidir.

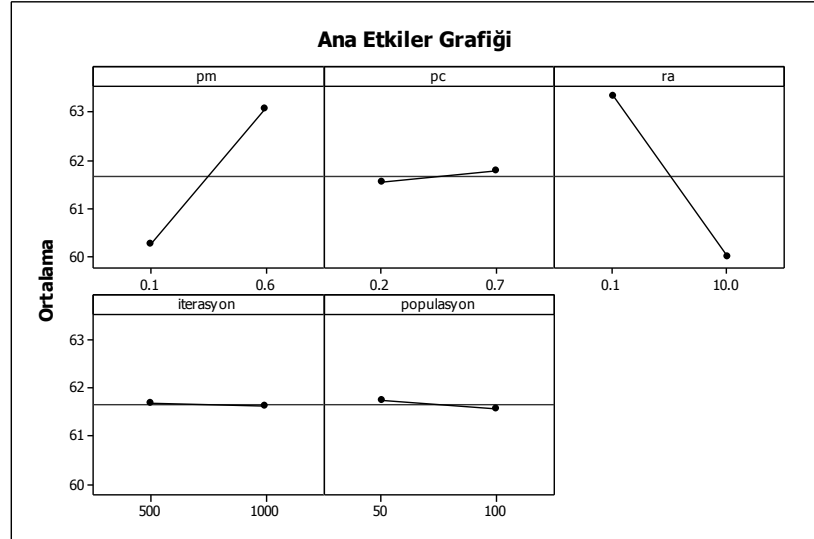
Tablo 6.16 SS algoritması Glass veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		61.652	0.1716	359.23	0	
Pm	2.791	1.396	0.1716	8.13	0	
Pc	0.24	0.12	0.1716	0.7	0.485	
Ra	-3.356	-1.678	0.1716	-9.78	0	
İterasyon	-0.086	-0.043	0.1716	-0.25	0.803	
Popülasyon	-0.154	-0.077	0.1716	-0.45	0.654	
pm*pc	-0.171	-0.086	0.1716	-0.5	0.618	
pm*ra	2.808	1.404	0.1716	8.18	0	
pm*iterasyon	0.223	0.111	0.1716	0.65	0.517	
pm*popülasyon	0.257	0.128	0.1716	0.75	0.455	
pc*ra	0.086	0.043	0.1716	0.25	0.803	
pc*iterasyon	-0.068	-0.034	0.1716	-0.2	0.842	
pc*popülasyon	0.205	0.103	0.1716	0.6	0.55	
ra*iterasyon	-0.137	-0.068	0.1716	-0.4	0.69	
ra*popülasyon	0	0	0.1716	0	1	
iterasyon*popülasyon	-0.428	-0.214	0.1716	-1.25	0.213	
pm*pc*ra	-0.291	-0.146	0.1716	-0.85	0.397	
pm*pc*iterasyon	0.205	0.103	0.1716	0.6	0.55	
pm*pc*popülasyon	-0.445	-0.223	0.1716	-1.3	0.196	
pm*ra*iterasyon	-0.479	-0.24	0.1716	-1.4	0.164	
pm*ra*popülasyon	0.103	0.051	0.1716	0.3	0.765	
pm*iterasyon*populasyon	-0.154	-0.077	0.1716	-0.45	0.654	
pc*ra*iterasyon	0.017	0.009	0.1716	0.05	0.96	
pc*ra*popülasyon	0.086	0.043	0.1716	0.25	0.803	
pc*iterasyon*populasyon	-0.342	-0.171	0.1716	-1	0.319	
ra*iterasyon*populasyon	-0.103	-0.051	0.1716	-0.3	0.765	
pm*pc*ra*iterasyon	0.394	0.197	0.1716	1.15	0.252	
pm*pc*ra*popülasyon	0.017	0.009	0.1716	0.05	0.96	
pm*pc*iterasyon*populasyon	0.103	0.051	0.1716	0.3	0.765	
pm*ra*iterasyon*populasyon	0.205	0.103	0.1716	0.6	0.55	
pc*ra*iterasyon*populasyon	0.051	0.026	0.1716	0.15	0.881	
pm*pc*ra*iterasyon*populasyon	0.051	0.026	0.1716	0.15	0.881	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	1531.41	1531.41	306.282	32.49	0
İkili_Etkileşimler	10	662.98	662.98	66.298	7.03	0
Üçlü_Etkileşimler	10	57.98	57.98	5.798	0.62	0.801
Dörtlü_Etkileşimler	5	16.87	16.87	3.373	0.36	0.877
Beşli_Etkileşimler	1	0.21	0.21	0.211	0.02	0.881
Artık_Hata	288	2714.58	2714.58	9.426		
Toplam	319	4984.03				

SS algoritması Glass veri kümesi için oluşturulan normal olasılık eğrisi ve pareto grafiği ekler bölümünde verilmiştir. Şekil 6.24’de gösterilen ana etkiler grafiği incelendiğinde parametrelerin etkin düzeyleri olarak aşağıdaki değerler elde edilmektedir.

- Pm: 0.1
- Pc: 0.2

- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

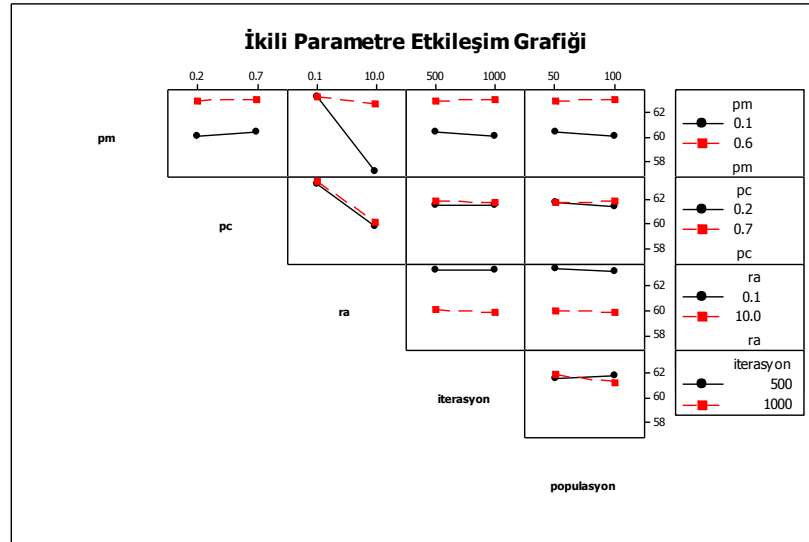


Şekil 6.24 SS algoritması Glass veri kümesi tekli parametre etkileri

Şekil 6.25’de verilen etkileşim grafiği üzerinden parametrelerin ikili etkileşimleri incelendiğinde parametre düzeyleri

- Pm: 0.1
- Pc: 0.2
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

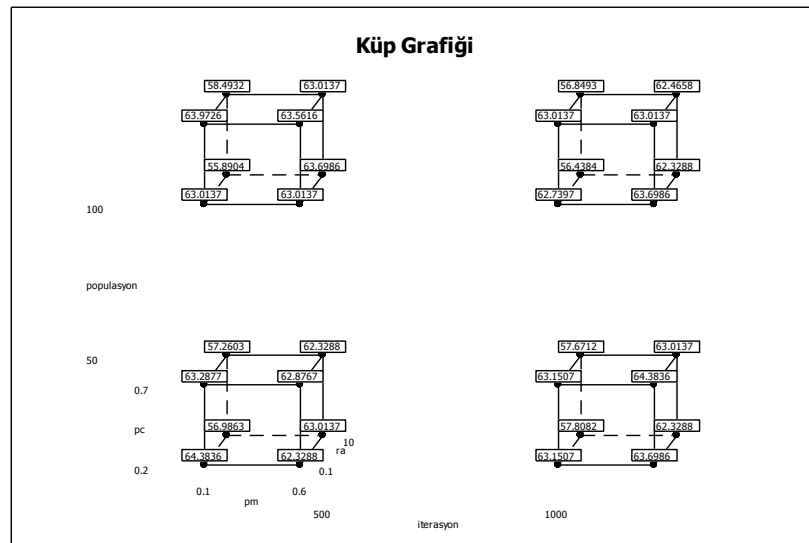
olarak belirlenmiştir.



Şekil 6.25 SS algoritması Glass veri kümesi ikili parametre etkileşimleri

Şekil 6.26’de verilen küp grafiği incelendiğinde parametrelerin ideal düzeyleri aşağıdaki şekilde belirlenmiştir.

- Pm: 0.1
- Pc: 0.2
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.26 SS algoritması Glass veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda Glass veri kümesi için en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.2
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Bu parametrelerden pm ve ra parametreleri ana etkiler ve etkileşimler bakımından etkili çıkmıştır ve yukarıda verilen değerlerinde kullanılmalıdır. Diğer parametreler ise herhangi bir düzeyinde kullanılabilir.

6.6.2.6 Soybean Veri Kümesi

Deney tasarımı sonuçları Tablo 6.17'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında pm, pc ve ra tek başlarına etkilidir. pm ile ra ve popülasyon büyüklüğü ile iterasyon sayısı ayrıca ikili etkileşimle çözüm üzerinde etkilidir. pc, ra ve iterasyon sayısı arasında ise üçlü etkileşim söz konusudur.

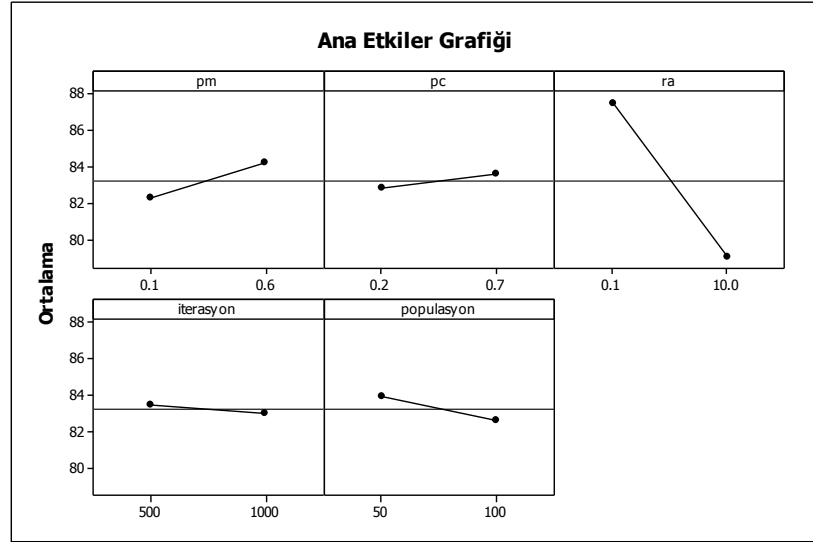
Tablo 6.17 SS algoritması Soybean veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		83.231	0.3255	255.68	0	
Pm	1.908	0.954	0.3255	2.93	0.004	
Pc	0.748	0.374	0.3255	1.15	0.252	
ra	-8.471	-4.235	0.3255	-13.01	0	
iterasyon	-0.48	-0.24	0.3255	-0.74	0.462	
populasyon	-1.306	-0.653	0.3255	-2.01	0.046	
pm*pc	-0.011	-0.006	0.3255	-0.02	0.986	
pm*ra	3.717	1.858	0.3255	5.71	0	
pm*iterasyon	0.056	0.028	0.3255	0.09	0.932	
pm*populasyon	0.48	0.24	0.3255	0.74	0.462	
pc*ra	-0.078	-0.039	0.3255	-0.12	0.905	
pc*iterasyon	-1.15	-0.575	0.3255	-1.77	0.079	
pc*populasyon	-1.172	-0.586	0.3255	-1.8	0.073	
ra*iterasyon	0.346	0.173	0.3255	0.53	0.596	
ra*populasyon	-0.033	-0.017	0.3255	-0.05	0.959	
iterasyon*populasyon	1.306	0.653	0.3255	2.01	0.046	
pm*pc*ra	0.815	0.407	0.3255	1.25	0.212	
pm*pc*iterasyon	0.859	0.43	0.3255	1.32	0.188	
pm*pc*populasyon	-0.234	-0.117	0.3255	-0.36	0.719	
pm*ra*iterasyon	-0.145	-0.073	0.3255	-0.22	0.824	
pm*ra*populasyon	-0.525	-0.262	0.3255	-0.81	0.421	
pm*iterasyon*populasyon	0.056	0.028	0.3255	0.09	0.932	
pc*ra*iterasyon	-1.752	-0.876	0.3255	-2.69	0.008	
pc*ra*populasyon	-0.346	-0.173	0.3255	-0.53	0.596	
pc*iterasyon*populasyon	0.77	0.385	0.3255	1.18	0.238	
ra*iterasyon*populasyon	0.658	0.329	0.3255	1.01	0.313	
pm*pc*ra*iterasyon	0.926	0.463	0.3255	1.42	0.156	
pm*pc*ra*populasyon	-0.703	-0.352	0.3255	-1.08	0.281	
pm*pc*iterasyon*populasyon	0.279	0.14	0.3255	0.43	0.669	
pm*ra*iterasyon*populasyon	0.257	0.128	0.3255	0.39	0.694	
pc*ra*iterasyon*populasyon	0.391	0.195	0.3255	0.6	0.549	
pm*pc*ra*iterasyon*populasyon	-0.056	-0.028	0.3255	-0.09	0.932	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	6231.6	6231.55	1246.31	36.75	0
İkili_Etkileşimler	10	1485.8	1485.83	148.58	4.38	0
Üçlü_Etkileşimler	10	477.9	477.86	47.79	1.41	0.175
Dörtlü_Etkileşimler	5	131.9	131.91	26.38	0.78	0.566
Beşli_Etkileşimler	1	0.2	0.25	0.25	0.01	0.932
Artık_Hata	288	9766.3	9766.26	33.91		
Toplam	319	18093.7				

SS algoritması Soybean veri kümesi için oluşturulan normal olasılık eğrisi ve pareto grafiği ekler bölümünde verilmiştir. Şekil 6.27 ana etkiler grafiğini göstermektedir. Ana etkiler grafiği incelendiğinde hatalı sınıflandırma oranını en küçükleyecek şekilde parametre düzeyleri aşağıdaki şekilde belirlenmiştir.

- Pm: 0.1

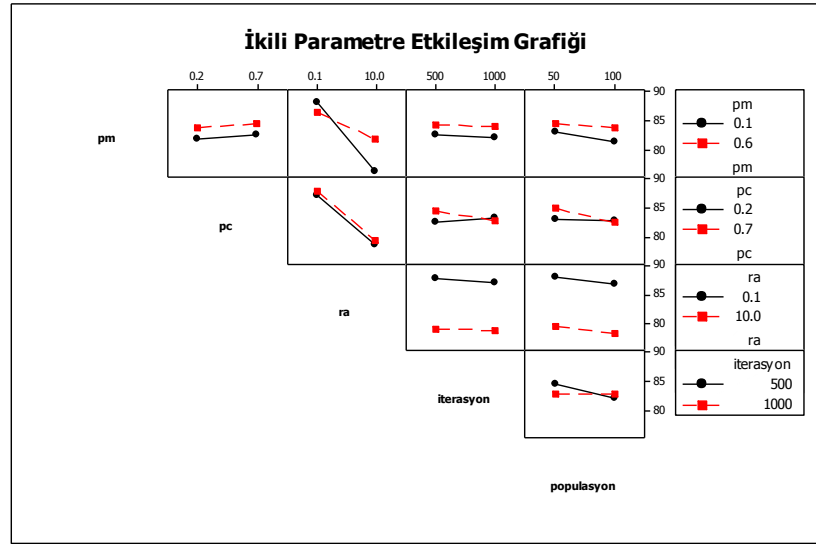
- Pc: 0.2
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.27 SS algoritması Soybean veri kümesi tekli parametre etkileri

Şekil 6.28’de etkileşim grafikleri gösterilmektedir. İkili etkileşimler dikkate alındığında parametre düzeyleri şu şekilde belirlenmiştir.

- Pm: 0.1
- Pc: 0.2 veya 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

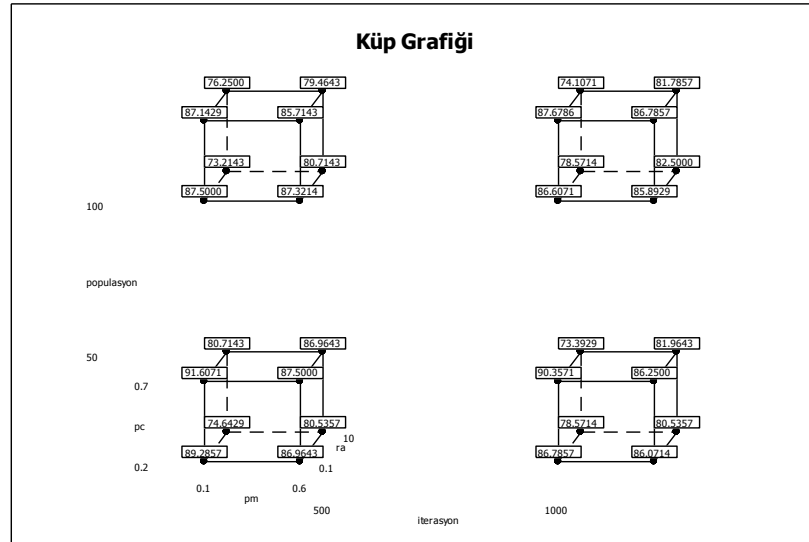


Şekil 6.28 SS algoritması Soybean veri kümesi ikili parametre etkileşimleri

Şekil 6.29’da gösterilen parametrelerin küp grafiği incelendiğinde ideal parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir.



Şekil 6.29 SS algoritması Soybean veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda en uygun parametre düzeyleri

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Soybean veri kümesinde tüm parametreler ana etkiler veya etkileşimler bakımından etkilidir. Dolayısıyla yukarıda verilen düzeylerinde kullanılmalılardır.

6.6.2.7 Thyroid Veri Kümesi

Deney tasarımı sonuçları Tablo 6.18’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında etkili olan bir parametre yoktur.

Tablo 6.18 SS algoritması Thyroid veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		7.2927	0.000129	56567.1	0	
pm	0	0	0.000129	0	1	
pc	0.00036	0.00018	0.000129	1.41	0.158	
ra	0.00036	0.00018	0.000129	1.41	0.158	
iterasyon	0	0	0.000129	0	1	
populasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
pm*pc	0	0	0.000129	0	1	
pm*ra	0	0	0.000129	0	1	
pm*iterasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
pm*populasyon	0	0	0.000129	0	1	
pc*ra	-0.00036	-0.00018	0.000129	-1.41	0.158	
pc*iterasyon	0	0	0.000129	0	1	
pc*populasyon	0.00036	0.00018	0.000129	1.41	0.158	
ra*iterasyon	0	0	0.000129	0	1	
ra*populasyon	0.00036	0.00018	0.000129	1.41	0.158	
iterasyon*populasyon	0	0	0.000129	0	1	
pm*pc*ra	0	0	0.000129	0	1	
pm*pc*iterasyon	0.00036	0.00018	0.000129	1.41	0.158	
pm*pc*populasyon	0	0	0.000129	0	1	
pm*ra*iterasyon	0.00036	0.00018	0.000129	1.41	0.158	
pm*ra*populasyon	0	0	0.000129	0	1	
pm*iterasyon*populasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
pc*ra*iterasyon	0	0	0.000129	0	1	
pc*ra*populasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
pc*iterasyon*populasyon	0	0	0.000129	0	1	
ra*iterasyon*populasyon	0	0	0.000129	0	1	
pm*pc*ra*iterasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
pm*pc*ra*populasyon	0	0	0.000129	0	1	
pm*pc*iterasyon*populasyon	0.00036	0.00018	0.000129	1.41	0.158	
pm*ra*iterasyon*populasyon	0.00036	0.00018	0.000129	1.41	0.158	
pc*ra*iterasyon*populasyon	0	0	0.000129	0	1	
pm*pc*ra*iterasyon*populasyon	-0.00036	-0.00018	0.000129	-1.41	0.158	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	5	3.19E-05	0.00003191	6.38E-06	1.2	0.309
İkili_Etkileşimler	10	4.26E-05	0.00004255	4.25E-06	0.8	0.629
Üçlü_Etkileşimler	10	4.26E-05	0.00004255	4.25E-06	0.8	0.629
Dörtlü_Etkileşimler	5	3.19E-05	0.00003191	6.38E-06	1.2	0.309
Beşli_Etkileşimler	1	1.06E-05	0.00001064	1.06E-05	2	0.158
Artık_Hata	288	0.001532	0.00153176	5.32E-06		
Toplam	319	0.001691				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir.

Yapılan deney tasarımı sonucunda etkili parametreleri dikkate alarak SS algoritması için tüm veri kümelerini kapsayacak parametre düzeyleri şu şekilde elde edilmiştir:

- Pm: 0.1
- Pc: 0.7
- Ra: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

6.6.3 Stokastik Fraktal Arama Algoritması Deney Tasarımı

Stokastik fraktal arama algoritmasında popülasyon büyüklüğü, iterasyon sayısı ve q olmak üzere üç parametre mevcuttur. Deneyde test edilen parametreler ve parametre düzeyleri Tablo 6.19’da gösterildiği gibidir.

Tablo 6.19 SFS algoritması deney tasarımı parametreleri

Parametreler	Değerler
Popülasyon büyüklüğü	50 – 100
İterasyon sayısı	500 – 1000
q	3 – 10

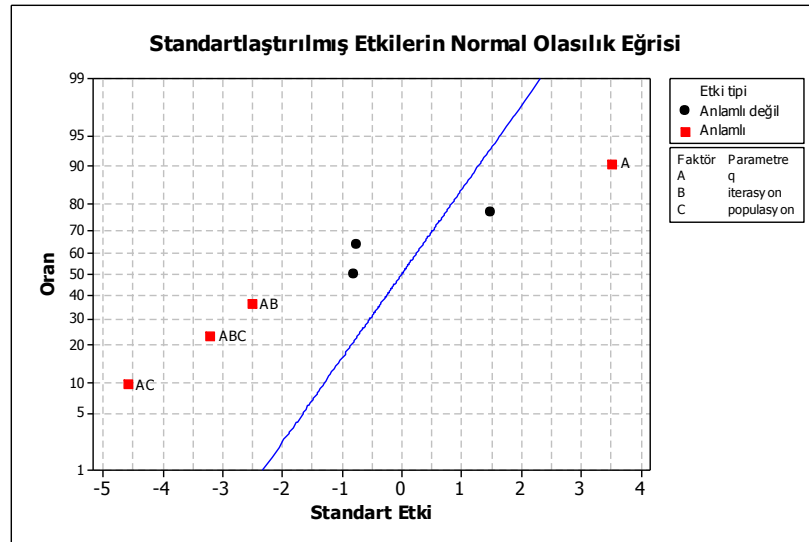
6.6.3.1 Gene Veri Kümesi

Deney tasarımı sonuçları Tablo 6.20’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında q parametresi tek başına etkilidir. İkili etkileşimler açısından q ile iterasyon sayısı arasında ve q ile popülasyon büyüklüğü arasında etkileşim söz konusudur. Ayrıca algoritmanın üç parametresi arasında da üçlü etkileşim vardır.

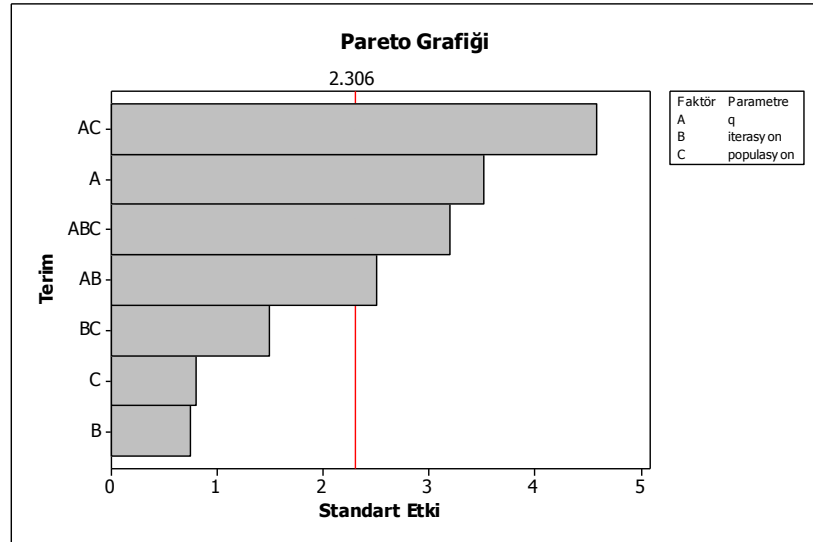
Tablo 6.20 SFS algoritması Gene veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		30.076	0.5917	50.83	0	
q	4.161	2.081	0.5917	3.52	0.008	
iterasyon	-0.883	-0.441	0.5917	-0.75	0.477	
populasyon	-0.946	-0.473	0.5917	-0.8	0.447	
q*iterasyon	-2.963	-1.482	0.5917	-2.5	0.037	
q*populasyon	-5.422	-2.711	0.5917	-4.58	0.002	
iterasyon*populasyon	1.765	0.883	0.5917	1.49	0.174	
q*iterasyon*populasyon	-3.783	-1.892	0.5917	-3.2	0.013	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	75.964	75.964	25.321	4.52	0.039
İkili_Etkileşimler	3	165.207	165.207	55.069	9.83	0.005
Üçlü_Etkileşimler	1	57.247	57.247	57.247	10.22	0.013
Artık_Hata	8	44.812	44.812	5.602		
Toplam	15	343.23				

Şekil 6.30'daki normal olasılık eğrisi ve Şekil 6.31'deki parato grafiği de Tablo 6.20'de elde edilen sonucu desteklemektedir.



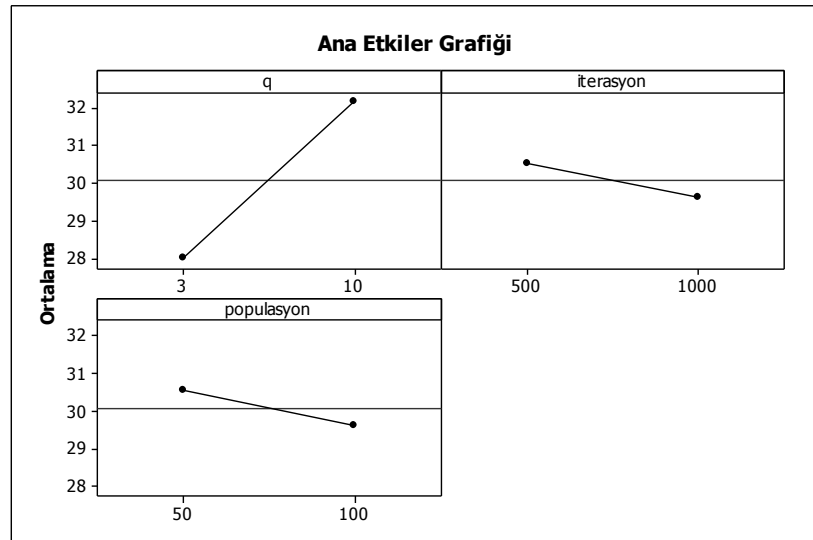
Şekil 6.30 SFS algoritması Gene veri kümesi normal olasılık eğrisi



Şekil 6.31 SFS algoritması Gene veri kümesi pareto grafiği

Şekil 6.32 ana etkiler grafiğini göstermektedir. Ana etkiler grafiğine göre parametre düzeyleri şu şekilde belirlenmiştir.

- q: 3
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100

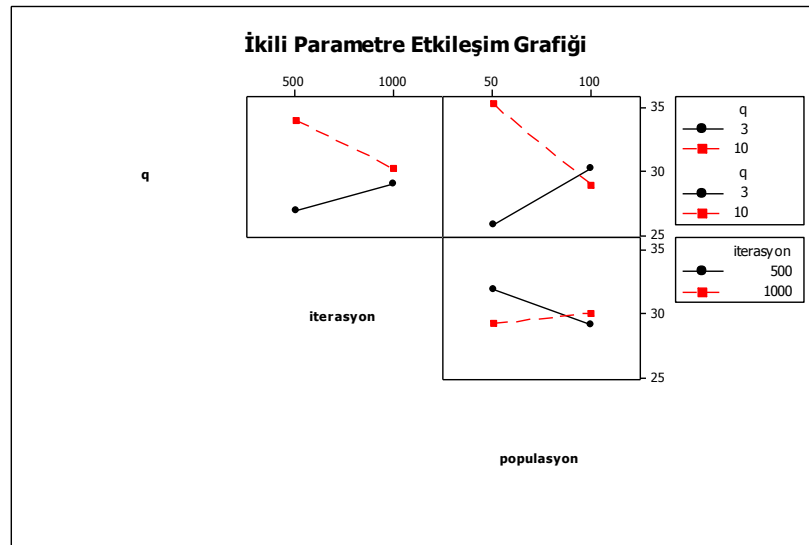


Şekil 6.32 SFS algoritması Gene veri kümesi tekli parametre etkileri

Şekil 6.33 etkileşim grafiğini göstermektedir. Grafiğe göre q ile populasyon büyüklüğü ve iterasyon sayısı ile populasyon büyüklüğü parametreleri arasında ikili etkileşim

olduğu görülmektedir. İkili etkileşimler dikkate alındığında parametre düzeyleri şu şekildedir:

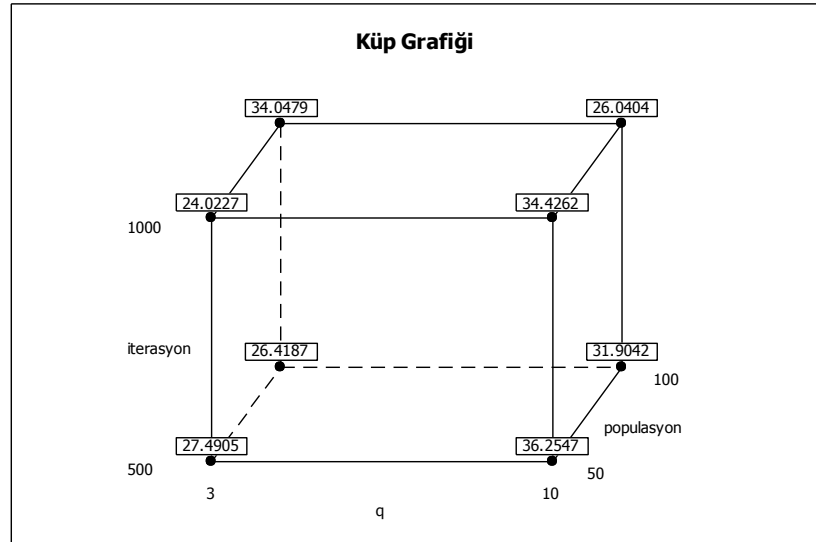
- q : 3
- İterasyon sayısı: 500
- Popülasyon büyüklüğü: 100



Şekil 6.33 SFS algoritması Gene veri kümesi ikili parametre etkileşimleri

Şekil 6.34 küp grafiğini göstermektedir. Küp grafiğe göre parametre düzeyleri şu şekildedir:

- q : 3
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 50



Şekil 6.34 SFS algoritması Gene veri kümesi küp grafiği

Yapılan bütün testler ve analizler sonucunda ele alınan veri kümesi için en uygun parametre düzeyleri

- q: 3
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 50

olarak belirlenmiştir. İlgili veri kümesinde tüm parametreler ana etkiler veya etkileşimler bakımından etkili olduğu için yukarıda verilen parametre düzeyleri tüm parametreler için kullanılmalıdır.

6.6.3.2 Cancer Veri Kümesi

Deney tasarımı sonuçları Tablo 6.21'deki gibidir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında etkili olan bir parametre yoktur.

Tablo 6.21 SFS algoritması Cancer veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		0.57471	0.1136	5.06	0.001	
q	0	0	0.1136	0	1	
iterasyon	-0.14368	-0.07184	0.1136	-0.63	0.545	
populasyon	0.43103	0.21552	0.1136	1.9	0.094	
q*iterasyon	0.14368	0.07184	0.1136	0.63	0.545	
q*populasyon	0.14368	0.07184	0.1136	0.63	0.545	
iterasyon*populasyon	0	0	0.1136	0	1	
q*iterasyon*populasyon	0	0	0.1136	0	1	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	0.82574	0.82574	0.27525	1.33	0.33
İkili_Etkileşimler	3	0.16515	0.16515	0.05505	0.27	0.848
Üçlü_Etkileşimler	1	0	0	0	*	*
Artık_Hata	8	1.65147	1.65147	0.20643		
Toplam	15	2.64236				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir.

6.6.3.3 Card Veri Kümesi

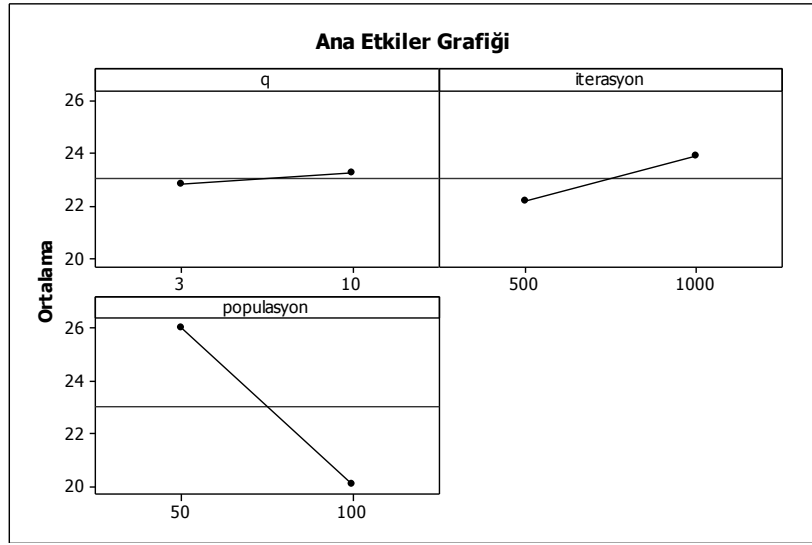
Deney tasarımı sonuçları Tablo 6.22’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında populasyon büyüklüğü parametresi tek başına etkilidir.

Tablo 6.22 SFS algoritması Card veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		23.038	1.158	19.89	0	
q	0.436	0.218	1.158	0.19	0.855	
iterasyon	1.744	0.872	1.158	0.75	0.473	
populasyon	-5.959	-2.98	1.158	-2.57	0.033	
q*iterasyon	-3.779	-1.89	1.158	-1.63	0.141	
q*populasyon	0.436	0.218	1.158	0.19	0.855	
iterasyon*populasyon	0	0	1.158	0	1	
q*iterasyon*populasyon	0.581	0.291	1.158	0.25	0.808	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	154.982	154.982	51.661	2.41	0.143
İkili_Etkileşimler	3	57.886	57.886	19.295	0.9	0.483
Üçlü_Etkileşimler	1	1.352	1.352	1.352	0.06	0.808
Artık_Hata	8	171.714	171.714	21.464		
Toplam	15	385.935				

Parametrelerin düzeylerine karar vermek için Şekil 6.35’de gösterilen ana etkiler grafiği incelendiğinde SFS algoritması Card veri kümesinde parametre düzeyleri aşağıdaki şekilde belirlenmiştir.

- q: 3
- İterasyon sayısı: 500
- Popülasyon büyüklüğü: 100



Şekil 6.35 SFS algoritması Card veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği, ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir.

Yapılan bütün testler ve analizler sonucunda ele alınan veri kümesi için en uygun parametre düzeyleri

- q: 3
- İterasyon sayısı: 500
- Popülasyon büyüklüğü: 100

olarak belirlenmiştir. Tasarım sonucunda Card veri kümesinde tek etkili parametre popülasyon büyüklüğüdür ve 100 düzeyi kullanılmalıdır. Diğer parametreler herhangi bir düzeyinde kullanılabilir.

6.6.3.4 Diabetes Veri Kümesi

Deney tasarımı sonuçları Tablo 6.23’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında etkili bir parametre çıkmamıştır.

Tablo 6.23 SFS algoritması Diabetes veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		26.3021	0.4143	63.48	0	
q	-1.5625	-0.7813	0.4143	-1.89	0.096	
iterasyon	-1.5625	-0.7812	0.4143	-1.89	0.096	
populasyon	-0.651	-0.3255	0.4143	-0.79	0.455	
q*iterasyon	1.0417	0.5208	0.4143	1.26	0.244	
q*populasyon	-1.6927	-0.8464	0.4143	-2.04	0.075	
iterasyon*populasyon	0.651	0.3255	0.4143	0.79	0.455	
q*iterasyon*populasyon	-0.3906	-0.1953	0.4143	-0.47	0.65	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	21.2267	21.2267	7.0756	2.58	0.127
İkili_Etkileşimler	3	17.4967	17.4967	5.8322	2.12	0.175
Üçlü_Etkileşimler	1	0.6104	0.6104	0.6104	0.22	0.65
Artık_Hata	8	21.9727	21.9727	2.7466		
Toplam	15	61.3064				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir. Etkili parametre çıkmadığı için parametreler bu veri kümesinde herhangi bir düzeyinde kullanılabilir.

6.6.3.5 Glass Veri Kümesi

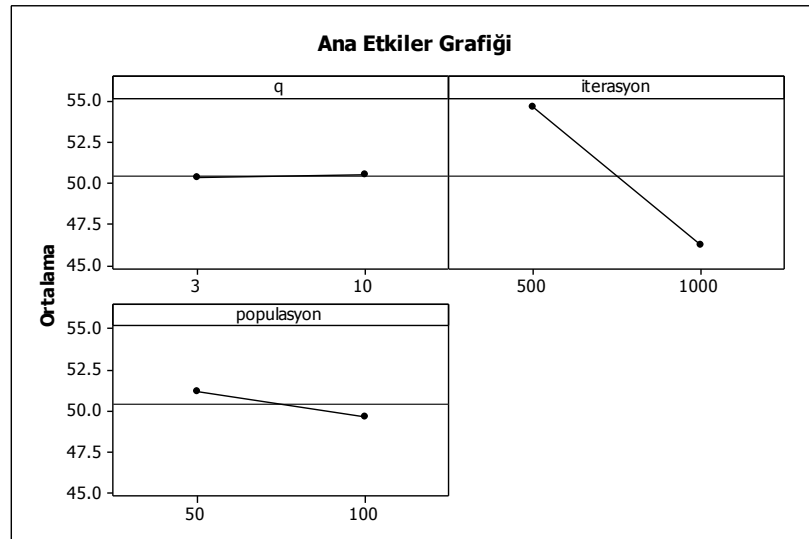
Deney tasarımı sonuçları Tablo 6.24’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında iterasyon sayısı parametresi tek başına etkilidir.

Tablo 6.24 SFS algoritması Glass veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		50.428	1.658	30.42	0	
q	0.171	0.086	1.658	0.05	0.96	
iterasyon	-8.39	-4.195	1.658	-2.53	0.035	
populasyon	-1.541	-0.771	1.658	-0.46	0.654	
q*iterasyon	-2.226	-1.113	1.658	-0.67	0.521	
q*populasyon	-0.171	-0.086	1.658	-0.05	0.96	
iterasyon*populasyon	0.171	0.086	1.658	0.05	0.96	
q*iterasyon*populasyon	4.281	2.14	1.658	1.29	0.233	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	291.21	291.21	97.071	2.21	0.165
İkili_Etkileşimler	3	20.06	20.06	6.685	0.15	0.925
Üçlü_Etkileşimler	1	73.3	73.3	73.302	1.67	0.233
Artık_Hata	8	351.85	351.85	43.981		
Toplam	15	736.42				

Şekil 6.36 ana etkiler grafiğini göstermektedir. Ana etkiler grafiğine göre q parametresinin düzeyleri arasında çok az farklılık görülmektedir ve parametre düzeyleri aşağıdaki şekilde belirlenmektedir.

- q: 3
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 100



Şekil 6.36 SFS algoritması Glass veri kümesi tekli parametre etkileri

Normal olasılık eğrisi, pareto grafiği, ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir.

Yapılan bütün testler ve analizler sonucunda ele alınan veri kümesi için en uygun parametre düzeyleri

- q: 10
- İterasyon sayısı: 1000
- Popülasyon büyüklüğü: 50

olarak belirlenmiştir. İlgili veri kümesinde sadece iterasyon sayısı parametresi etkili olduğu için yukarıda verilen düzeyi kullanılmalıdır. Diğer iki parametre herhangi bir düzeyinde kullanılabilir.

6.6.3.6 Soybean Veri Kümesi

Deney tasarımı sonuçları Tablo 6.25’de verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında etkili parametre yoktur.

Tablo 6.25 SFS algoritması Soybean veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		62.612	1.761	35.55	0	
q	5.134	2.567	1.761	1.46	0.183	
iterasyon	5.58	2.79	1.761	1.58	0.152	
populasyon	-1.562	-0.781	1.761	-0.44	0.669	
q*iterasyon	5.134	2.567	1.761	1.46	0.183	
q*populasyon	-0.223	-0.112	1.761	-0.06	0.951	
iterasyon*populasyon	-2.455	-1.228	1.761	-0.7	0.505	
q*iterasyon*populasyon	0.67	0.335	1.761	0.19	0.854	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	239.756	239.756	79.919	1.61	0.262
İkili_Etkileşimler	3	129.743	129.743	43.248	0.87	0.495
Üçlü_Etkileşimler	1	1.794	1.794	1.794	0.04	0.854
Artık_Hata	8	397.003	397.003	49.625		
Toplam	15	768.296				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir. İlgili veri kümesinde hiçbir parametre etkili çıkmadığı için parametreler herhangi bir düzeyinde kullanılabilir.

6.6.3.7 Thyroid Veri Kümesi

Deney tasarımı sonuçları Tablo 6.26’da verilmiştir. Anlamlılık düzeyi $\alpha = 0.05$ olarak dikkate alındığında hiçbir parametre etkili çıkmamıştır.

Tablo 6.26 SFS algoritması Thyroid veri kümesi deney tasarımı sonuçları

Çözüm İçin Tahmin Edilen Etki ve Katsayılar						
Terim	Etki	Katsayı	SE_Katsayı	T	P	
Sabit		4.919	0.3391	14.51	0	
q	0.3282	0.1641	0.3391	0.48	0.641	
iterasyon	-1.0939	-0.547	0.3391	-1.61	0.145	
populasyon	-0.5761	-0.2881	0.3391	-0.85	0.42	
q*iterasyon	-0.1896	-0.0948	0.3391	-0.28	0.787	
q*populasyon	-0.1969	-0.0985	0.3391	-0.29	0.779	
iterasyon*populasyon	0.3646	0.1823	0.3391	0.54	0.605	
q*iterasyon*populasyon	-1.2106	-0.6053	0.3391	-1.79	0.112	
Çözüm İçin Varyans Analizi						
Kaynak	DF	SeqSS	AdjSS	AdjMS	F	P
Ana_Etkiler	3	6.5453	6.5453	2.1818	1.19	0.374
İkili_Etkileşimler	3	0.8308	0.8308	0.2769	0.15	0.926
Üçlü_Etkileşimler	1	5.8624	5.8624	5.8624	3.19	0.112
Artık_Hata	8	14.7143	14.7143	1.8393		
Toplam	15	27.9527				

Normal olasılık eğrisi, pareto grafiği, ana etkiler ve ikili etkileşim grafikleri ve küp grafiği ekler bölümünde verilmiştir. İlgili veri kümesinde hiçbir parametre etkili çıkmadığı için parametrelerin herhangi bir düzeyi kullanılabilir.

SFS algoritmasında diğer iki algoritmada olduğu gibi tek bir parametre kümesi elde edilememiştir. Etkili veya etkileşimli olan parametreler açısından incelendiğinde populasyon büyüklüğü parametresinin düzeyi veri kümesine göre farklılaşmaktadır. Dolayısıyla populasyon büyüklüğü parametresi veri kümesine göre 50 veya 100 değerinin birinde, iterasyon sayısı 1000 düzeyinde ve q parametresi 3 düzeyinde kullanılmalıdır.

6.7 Yeni Metasezgisel Algoritmaların Parametre Değerleri

Bölüm 6.6’da yapılan deneysel çalışmalar sonucunda elde edilen iyon hareketi algoritması parametre değerleri Tablo 6.27’de gösterilmiştir. Çalışmada ele alınan tüm literatür verileri tabloda verilen parametre değerleri ile çalıştırılmıştır.

Tablo 6.27 IM algoritması parametre değerleri

Parametre	Popülasyon büyüklüğü	İterasyon sayısı
Değer	100	1000

Deney tasarımı sonucunda belirlenen sosyal örümcek ağı parametre değerleri Tablo 6.28’de gösterilmiştir. IM algoritmasında olduğu gibi tüm veri kümeleri tabloda verilen parametre değerleri ile çalıştırılmıştır.

Tablo 6.28 SS algoritması parametre değerleri

Parametre	Popülasyon büyüklüğü	İterasyon sayısı	pm	pc	ra
Değer	100	1000	0.1	0.7	10

Stokastik fraktal arama algoritması için deney tasarımı sonucunda diğer algoritmalarda olduğu gibi tek bir parametre kümesi elde edilememiştir. Popülasyon büyüklüğü parametresi diğer veri kümelerinden farklı olarak 50 belirlenmiş, diğerleri için ise parametre düzeyi olarak 100 belirlenmiştir. Bu farklılığı belirtmek için SFS algoritmasının parametre değerlerini gösteren Tablo 6.29’da popülasyon büyüklüğü değeri olarak 2 değer gösterilmektedir. Gene veri kümesi hariç diğer veri kümeleri aynı parametre kümesi ile çalıştırılmış, Gene veri kümesinde ise popülasyon büyüklüğü değeri olarak 100 yerine 50 kullanılmıştır.

Tablo 6.29 SFS algoritması parametre değerleri

Parametre	Popülasyon büyüklüğü	İterasyon Sayısı	q
Değer	50 / 100	1000	3

6.8 Yeni Metasezgiseller ile Elde Edilen Sonuçlar

İyon hareketi algoritması, sosyal örümcek algoritması ve stokastik fraktal arama algoritması sırasıyla Tablo 6.27, Tablo 6.28 ve Tablo 6.29’da verilen parametre değerleriyle yapay sinir ağlarının eğitimi için 30 kez çalıştırılmıştır.

Algoritmaların literatür veri kümeleri üzerinde elde ettikleri sonuçlar iki aşamada değerlendirilmiştir. İlk aşamada yeni nesil metasezgisel algoritmalar kendi aralarında karşılaştırılmış, hatalı sınıflandırma oranı ve çalışma süreleri açısından değerlendirilmişlerdir.

IM, SS ve SFS algoritmaları ile elde edilen hatalı sınıflandırma oranları (CEP) ve süreler (saniye) standart sapmaları ile birlikte Tablo 6.30'da gösterilmiştir. Tablodaki koyu işaretli değerler veri kümeleri için elde edilen en iyi değerleri ifade etmektedir.

Tablo 6.30 IM, SS ve SFS algoritmalarının elde ettiği sonuçlar

Veri kümesi	Etkinlik ölçütü	IM	SS	SFS
Cancer	Ort CEP	0.48	0.88	0.96
	Std. Sapma	0.26	0.51	1.26
	Ort. Süre	34.23	32.12	429.35
	Std. Sapma	1.53	0.75	22.64
Card	Ort CEP	21.26	27.42	25.12
	Std. Sapma	2.69	1.08	3.35
	Ort. Süre	56.86	49.27	603.65
	Std. Sapma	5.90	2.58	139.03
Diabetes	Ort CEP	24.79	25.07	27.34
	Std. Sapma	0.86	1.56	1.48
	Ort. Süre	54.64	36.54	480.83
	Std. Sapma	6.37	0.78	41.88
Gene	Ort CEP	20.30	45.87	36.37
	Std. Sapma	4.72	2.69	4.91
	Ort. Süre	351.34	358.03	4434.26
	Std. Sapma	6.33	4.05	78.38
Glass	Ort CEP	40.51	47.58	54.89
	Std. Sapma	3.55	5.54	6.89
	Ort. Süre	13.21	17.15	156.99
	Std. Sapma	0.57	0.35	4.94
Soybean	Ort CEP	45.71	73.93	64.76
	Std. Sapma	5.71	8.10	6.27
	Ort. Süre	82.43	99.97	911.49
	Std. Sapma	4.44	4.55	9.80
Thyroid	Ort CEP	3.71	7.29	6.13
	Std. Sapma	0.81	0.00	1.17
	Ort. Süre	349.17	287.39	3535.54
	Std. Sapma	14.96	1.48	53.15

Tablo 6.29 incelendiğinde bütün veri kümelerinde en küçük CEP değerini IM algoritmasının verdiği görülmektedir. Süre bakımından ise IM ile SS algoritmaları birbirlerine çok yakındır. Ayrıca SS ve SFS algoritmaları CEP değerleri bakımından birbirlerine yakındır fakat süre olarak SFS algoritması diğerlerinden çok geridedir.

IM, SS ve SFS algoritmaları ile elde edilen CEP ve süre değerleri arasındaki farklılıkların istatistiksel olarak anlamlı olup olmadıklarını test etmek için istatistiksel

hipotez testleri oluşturulmuştur. Her bir algoritma her bir veri kümesinde 30 kez çalıştırıldığı için merkezi limit teoreminden elde edilen CEP ve sürelerin normal dağılıma uyduğu varsayılabilir. Algoritmalar aynı eğitim ve test kümeleriyle çalıştırıldığından ve normal dağılım varsayıldığından dolayı istatistiksel analiz testi olarak Eşleştirmeli-t testi kullanılmıştır. Tablo 6.31 $\alpha=0.05$ anlamlılık düzeyinde Eşleştirmeli-t testinden elde edilen p değerlerini göstermektedir. Tabloda koyu işaretlenen değerler karşılaştırılan iki algoritmanın CEP/süre değerleri arasında istatistiksel olarak anlamlı bir farklılık olduğunu ($p<0.05$) ifade etmektedir.

Tablo 6.31 Eşleştirmeli-t testi p değerleri

		Cancer	Card	Diabetes	Gene	Glass	Soybean	Thyroid
IM / SS	CEP	0.000	0.000	0.193	0.000	0.000	0.000	0.000
	Süre	1.000	1.000	0.000	0.000	0.000	0.000	1.000
IM / SFS	CEP	0.026	0.000	0.000	0.000	0.000	0.000	0.000
	Süre	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SS / SFS	CEP	0.382	1.000	0.955	1.000	0.714	1.000	1.000
	Süre	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Tablo 6.31'e göre IM algoritması CEP değerleri ile SS algoritması CEP değerleri arasında Diabetes veri kümesi hariç diğer veri kümelerinin hepsinde istatistiksel olarak anlamlı bir farklılık vardır ve IM algoritmasının CEP değerleri daha düşüktür. IM algoritması ile SFS algoritması CEP değerleri karşılaştırıldığında ise bütün veri kümelerinde IM algoritmasının bulduğu CEP değerleri daha iyidir ve SFS algoritması sonuçları ile aralarında istatistiksel olarak anlamlı bir farklılık vardır. Süre bakımından ise IM algoritması bazı veri kümelerinde SS algoritmasını ve bütün veri kümelerinde SFS algoritmasını geçmiştir. SS algoritması ile SFS algoritması karşılaştırıldığında ise, CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık olmadığı görülmektedir. Süre bakımından SS algoritması bütün veri kümelerinde SFS algoritmasından istatistiksel olarak iyi durumdadır.

İkinci aşamada IM, SS ve SFS algoritmaları geleneksel gradyan-tabanlı yöntemlerle ve populasyon-tabanlı metasezgisel algoritmalarla karşılaştırılmıştır. Karşılaştırılan algoritmalarından geri yayılım algoritması (BP) ve Levenberg-Marquardt (LM) gradyan-tabanlı, genetik algoritmalar (GA), parçacık sürü optimizasyonu (PSO), diferansiyel gelişim algoritması (DE), yapay arı kolonisi algoritması (ABC), harmoni algoritması (HA) ve av arama algoritması (HS) populasyon tabanlı algoritmalar. Bu algoritmalar

ile ilgili sonuçlar Kulluk [36] ve Karaboğa ve Öztürk'ün [35] çalışmalarından alınmıştır. Karşılaştırmaların anlamlı olması açısından bu çalışmalarda kullanılan YSA yapısı ve veri kümesinin aynı oranlarda basit bölümlere yöntemi ile eğitim ve test kümelerine parçalanması ele alınmıştır. Tablo 6.32 karşılaştırma sonuçlarını göstermektedir. Tabloda tüm veri kümeleri için ortalama hatalı sınıflandırma oranı ve standart sapmalar gösterilmektedir. Tablodaki koyu işaretli değerler veri kümeleri için elde edilen en iyi değerleri ifade etmektedir.

Tablo 6.32 Karşılaştırma Sonuçları

		Cancer	Card	Diabetes	Gene	Glass	Soybean	Thyroid
ABC	Ort. CEP	1.14	13.53	25.22	29.5	45.62	38.63	6.95
	St. Sapma	0.00	1.17	0.97	1.18	3.11	3.18	0.01
PSO	Ort. CEP	2.01	15.58	27.5	36.3	52.49	60.5	7.27
	St. Sapma	0.86	1.59	2.05	2.1	7.14	8.5	0.02
DE	Ort. CEP	1.19	13.9	24.84	31.18	44.9	74.9	7.08
	St. Sapma	0.28	1.26	1.32	2.18	2.84	2.1	0.95
GA	Ort. CEP	1.35	13.56	26.51	31.02	50.18	40.39	7.09
	St. Sapma	0.28	1.21	1.21	0.71	3.15	4.93	0.03
BP	Ort. CEP	1.89	13.86	28.27	11.37	59.09	61.16	7.26
	St. Sapma	0.43	0.47	6.3	1.15	9.52	19.18	0
LM	Ort. CEP	8.18	21.59	29.8	14.35	42.99	25.51	2.61
	St. Sapma	10.29	7.98	3.77	2.48	11.55	9.89	1.81
HA	Ort. CEP	2.53	15.17	31.67	14.23	58.24	71.31	7.28
	St. Sapma	0.39	1.11	0.73	1.33	2.36	4.29	0
HS	Ort. CEP	6.49	17.15	36.46	21.52	42.89	69.29	7.28
	St. Sapma	1.75	0.92	0.00	4.43	6.55	6.82	0
IM	Ort. CEP	0.48	21.26	24.79	20.30	40.51	45.71	3.71
	St. Sapma	0.26	2.69	0.86	4.72	3.55	5.71	0.81
SS	Ort. CEP	0.88	27.42	25.07	45.87	47.58	73.93	7.29
	St. Sapma	0.51	1.08	1.56	2.69	5.54	8.10	0.00
SFS	Ort. CEP	0.96	25.12	27.34	36.37	54.89	64.76	6.13
	St. Sapma	1.26	3.35	1.48	4.91	6.89	6.27	1.17

Tablo 6.32 incelendiğinde IM algoritmasının ele alınan yedi veri kümesinden Cancer, Diabetes ve Glass veri kümelerinde en iyi sonucu verdiği görülmektedir. SS ve SFS algoritmalarının ise en iyi durumda olduğu bir veri kümesi bulunmamaktadır. Karşılaştırılan algoritmalarından ABC algoritması Card veri kümesinde, BP algoritması Gene veri kümesinde ve LM algoritması Soybean ve Thyroid veri kümelerinde en düşük CEP değerlerini vermiştir. Tüm algoritmalar toplu değerlendirildiğinde en fazla veri kümesinde en iyi değerleri veren algoritma IM algoritmasıdır.

Tablo 6.32’de sunulan algoritmaların ortalama hatalı sınıflandırma oranları arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını belirlemek için istatistiksel analiz gerçekleştirilmiştir. Karşılaştırmada kullanılan algoritmalar literatürde mevcut çalışmalardan alındığı için ve sadece ortalama değerleri mevcut olduğundan analiz tekniği olarak 1-örnek t testi uygulanmıştır. Böylece çalışmada ele alınan yeni metasezgisel algoritmaların 30 çalışması ile karşılaştırılan algoritmaların ortalama CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık olup olmadığı belirlenmiştir. Tablo 6.33 ortalama CEP değerleri için 1-örnek t-testi p değerlerini göstermektedir. Tabloda koyu işaretli değerler karşılaştırılan algoritmaların CEP değerleri arasında anlamlı bir farklılık olduğunu ifade etmektedir ($p < 0.05$). IM, SS ve SFS algoritmaları bir önceki aşamada kendi aralarında istatistiksel olarak değerlendirildiklerinden dolayı bu aşamada tekrar birbirleriyle karşılaştırmalı olarak değerlendirilmemişlerdir.

Tablo 6.33 Ortalama CEP değerleri için 1-örnek t testi p değerleri

	Cancer	Card	Diabetes	Gene	Glass	Soybean	Thyroid
IM/ABC	0.000	1.000	0.005	0.000	0.000	1.000	0.000
IM/PSO	0.000	1.000	0.000	0.000	0.000	0.000	0.000
IM/DE	0.000	1.000	0.369	0.000	0.000	0.000	0.000
IM/GA	0.000	1.000	0.000	0.000	0.009	1.000	0.000
IM/BP	0.000	1.000	0.000	1.000	0.000	0.000	0.000
IM/LM	0.000	0.257	0.000	1.000	0.000	1.000	1.000
IM/HA	0.000	1.000	0.000	1.000	0.000	0.000	0.000
IM/HS	0.000	1.000	0.000	0.087	0.001	0.000	0.000
SS/ABC	0.005	1.000	1.000	1.000	1.000	1.000	*
SS/PSO	0.000	1.000	0.000	1.000	1.000	1.000	*
SS/DE	0.001	1.000	1.000	1.000	1.000	0.262	*
SS/GA	0.000	1.000	1.000	1.000	1.000	1.000	*
SS/BP	0.000	1.000	0.000	1.000	0.000	1.000	*
SS/LM	0.000	1.000	0.000	1.000	1.000	1.000	*
SS/HA	0.000	1.000	0.000	1.000	0.000	0.954	*
SS/HS	0.000	1.000	0.000	1.000	1.000	0.998	*
SFS/ABC	0.222	1.000	1.000	1.000	1.000	1.000	0.000
SFS/PSO	0.000	1.000	0.287	0.530	0.964	1.000	0.000
SFS/DE	0.165	1.000	1.000	1.000	1.000	0.000	0.000
SFS/GA	0.053	1.000	0.998	1.000	1.000	1.000	0.000
SFS/BP	0.000	1.000	0.001	1.000	0.001	0.998	0.000
SFS/LM	0.000	1.000	0.000	1.000	1.000	1.000	1.000
SFS/HA	0.000	1.000	0.000	1.000	0.007	0.000	0.000
SFS/HS	0.000	1.000	0.000	1.000	1.000	0.000	0.000

Tablo 6.33 incelendiğinde Thyroid veri kümesi için SS algoritması ile karşılaştırılan algoritmalar için p değerlerinin hesaplanamadığı görülmektedir. Bunun nedeni Thyroid

veri kümesinde SS algoritmasının 30 çalıştırmada da aynı CEP değerini vermesi yani standart sapmasının “0” olmasıdır.

Tablodan görüldüğü gibi Cancer veri kümesinde IM algoritması ile karşılaştırmaya alınan diğer tüm algoritmaların CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık vardır ve IM algoritması daha iyi sonuç vermiştir. Aynı veri kümesinde SS algoritması ile karşılaştırmada dikkate alınan algoritmalar arasındaki p değerleri incelendiğinde SS algoritması ile tüm algoritmaların CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık olduğu görülmektedir ve SS algoritması karşılaştırmada dikkate alınan algoritmalarından daha düşük CEP değerleri vermiştir. Cancer veri kümesinde SFS algoritması ile karşılaştırılan algoritmaların p değerlerine göre BP, LM, HA ve HS algoritmaları CEP değerleri ile SFS algoritmaları CEP değerleri arasında anlamlı bir farklılık olduğu görülmektedir. SFS algoritması BP, LM, HA ve HS algoritmalarından daha iyi CEP değeri vermiştir.

Tablo 6.33’de verilen p değerlerine göre Card veri kümesinde IM, SS ve SFS algoritmaları ile karşılaştırmada dikkate alınan algoritmaların CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık yoktur.

IM algoritmasının karşılaştırılan algoritmalarından daha düşük CEP değeri verdiği Diabetes veri kümesinde ABC ve DE algoritmaları hariç diğer tüm karşılaştırılan algoritmaların CEP değerleri ile IM CEP değerleri arasında anlamlı bir farklılık olduğu görülmektedir, IM algoritması en düşük CEP değerini vermiştir. Aynı veri kümesinde SS algoritmasıyla elde edilen CEP değerleri ile PSO, BP, LM, HA ve HS algoritmaların CEP değerleri arasında istatistiksel olarak farklılık vardır ve SS algoritması daha iyi sonuçlar vermiştir. Aynı şekilde SFS algoritması da BP, LM, HA ve HS algoritmalarından daha düşük CEP değerleri vermiştir ve karşılaştırılan algoritmaların CEP değerleri arasındaki farklılık istatistiksel olarak anlamlıdır.

Gene veri kümesinde sadece IM algoritması ile ABC, PSO, DE ve GA algoritmalarının CEP değerleri arasında farklılık vardır ve IM algoritması diğer algoritmalarından daha düşük değer vermiştir. Gene veri kümesinde en düşük CEP değerini veren BP algoritması ile IM algoritmasının CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık olmadığı görülmektedir.

Yine en düşük CEP değerini IM algoritmasını verdiği Glass veri kümesinde de elde edilen CEP değerleri arasında istatistiksel olarak anlamlı bir farklılık vardır ve IM algoritması daha iyi durumdadır. SS ve SFS algoritmaların verdiği CEP değerleri BP ve HA algoritmalarıyla elde edilen CEP değerlerinden daha düşüktür ve aralarında anlamlı bir farklılık vardır.

Soybean veri kümesinde IM algoritması PSO, DE, BP, HA ve HS algoritmalarının CEP değerlerinden istatistiksel olarak anlamlı daha düşük CEP değerleri elde etmiştir. SFS algoritmasının ise DE, HA ve HS algoritmaların CEP değerlerinden daha iyi değerlerle çözüm üretebildiği görülmektedir.

Thyroid veri kümesinde IM ve SFS algoritmaları LM hariç diğer tüm karşılaştırılan algoritmalarından istatistiksel olarak anlamlı farklılıkla daha düşük CEP değerleri elde etmiştir. Bu veri kümesinde en iyi sonucu LM algoritması vermiştir.

Sonuç olarak IM algoritması karşılaştırılan tüm algoritmalarla göre genel anlamda en iyi sonuçları vermiştir. SS ve SFS algoritmaları ise diğer algoritmalar ile rekabet edebilecek düzeydedir.

6.9 Gerçek Hayat Verisi ile Yeni Nesil Metasezgisel Algoritmaların Testi

Tez çalışmasında ele alınan yeni nesil metasezgisel algoritmalar, literatürde yer alan veri kümeleri dışında bir gerçek hayat verisiyle de test edilmiştir. Bu gerçek hayat verisi Kayseri’de üretim yapan bir yay fabrikasından temin edilmiştir. Veri kümesi haftanın günleri, vardiyalar ve üretimi gerçekleştiren personel açısından üretilen ürün miktarını içermektedir. Dolayısıyla veri kümesinde haftanın günü, vardiya ve üretim personeli içeren 3 adet girdi değişkeni bulunmaktadır. Çıktı değişkeni yani hedef değişken olarak ise üretim miktarı ele alınmıştır. Veri kümesinde toplam 820 adet örnek mevcuttur ve eksik nitelik değeri bulunmamaktadır. Fabrika haftanın 7 günü 3 vardiya halinde çalışmaktadır. Çalışmanın amacı bir yay makinesinin haftanın günü, vardiya ve üretim personeli açısından üretim miktarını tahminlemektir.

Çalışmada ele alınan literatür verilerinden farklı olarak gerçek hayat verisinde hedef değişken kesikli değil sürekli. Dolayısıyla yapılan çalışma literatür verileri için

gerçekleştirilen sınıflandırma çalışmasından ziyade bir fonksiyon tahminleme çalışmasıdır.

Veri kümesini oluşturan girdi değişkenleri ve çıktı değişkeni Tablo 6.34’de gösterilmiştir.

Tablo 6.34 Üretim veri kümesi değişken tanımları

Değişken Adı	Değişken Tipi	Açıklama	Değerler
Gün	Kesikli	Haftanın günleri	{1, 2, 3, 4, 5, 6, 7}
Vardiya	Kesikli	Vardiyalar	{Sabah, Gündüz, Gece}
Personel	Kesikli	Üretimden sorumlu kişi	{1, 2, 3}
Üretim	Sürekli	Üretim miktarı (adet)	[672, 144 708]

Elde edilen sonuçların standartlığı ve anlamlılığı açısından gerçek hayat verisinde literatür verilerine uygulanan basit bölümlleme yerine, 10-katlı çapraz-geçerlilik prosedürü veri kümesini eğitim ve test kümelerine parçalamakta kullanılmıştır. Literatür verilerine 10-katlı çapraz-geçerlilik testinin uygulanamamasının sebebi, karşılaştırmada kullanılan algoritmalarda basit bölümlleme kullanılmış olması ve aynı yöntemi uygulayarak karşılaştırmaların anlamlı olmasının sağlanmak istenmesidir.

10-katlı çapraz-geçerlilik testinde veri kümesi 10 eşit alt kümeye ayrılır ve ele alınan algoritma her alt küme için bir kez çalıştırılır. Çalıştırmaların her seferinde farklı bir alt küme test kümesi, kalan 9 alt küme ise gruplandırılarak eğitim kümesi olarak ele alınır. Sınıflandırma veya fonksiyon tahminleme çalışmalarında eğitim kümesi algoritmanın iyi öğrenme kabiliyeti değerlendirmek, test kümesi ise algoritmanın genelleme yeteneğini ölçmek için kullanılır. Çalışmada ele alınan algoritmalar her bir alt küme için 3 kez çalıştırılmış, dolayısıyla toplamda üretim veri kümesi için her bir algoritmada 30 çalıştırma elde edilmiştir.

IM, SS ve SFS algoritmalarının YSA eğitiminde, eğitim ve genelleme kabiliyetlerini analiz etmek ve karşılaştırmak için ortalama kareli hatanın kökü (RMSE) performans ölçütü dikkate alınmıştır. Üretim verisi için gizli katmanda 6 nöron içeren ve sigmoid aktivasyon fonksiyonu kullanan 3-katlı ileri beslemeli bir YSA yapısı tasarlanmıştır.

Gerçek hayat verisinde de IM, SS ve SFS algoritmalarının parametre değerleri olarak deney tasarımı sonucu belirlenen ve Tablo 6.27, Tablo 6.28 ve Tablo 6.29’da verilen parametre değerleri kullanılmıştır.

Yeni nesil metasezgisellerle eğitilen ve test edilen üretim verisi sonuçları Tablo 6.35’de gösterilmiştir. Tabloda verilen değerler algoritmaların 30 çalıştırmasının ortalamasıdır. Ayrıca standart sapmalar da aynı tabloda verilmiştir. Tablo incelendiğinde 3 yeni nesil metasezgisel algoritmanın da üretim verisi üzerinde yakın hata değerlerine sahip olduğu görülmektedir. Ancak en düşük hata değerini iyon hareketi algoritması daha düşük standart sapma ile vermiştir. Literatür verilerinde sınıflandırma kabiliyetinde olduğu gibi üretim verisinde de iyon hareketi algoritması diğer metasezgisel algoritmalarından daha iyi tahminleme kabiliyetine sahiptir.

Tablo 6.35 Algoritmaların üretim verisi üzerindeki test sonuçları

Algoritma	RMSE	Standart sapma
IM	18107.73	65.64
SS	18252.86	579.62
SFS	18154.37	85.03

Algoritmaların elde ettiği hata değerleri arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını test etmek için istatistiksel analiz gerçekleştirilmiştir. Merkezi limit teoremindeki örneklemden örnek sayısı 30 veya üzerinde olduğunda normal dağılım söz konusudur varsayımını dikkate alınarak, gruplar bağımlı olduğu ve algoritmalar 30’ar kere çalıştırıldığı için Eşleştirilmiş-t testi istatistiksel analiz için kullanılmıştır. Hipotez olarak, ele alınan ilk algoritmanın ortalama hata değerlerinin ikinci algoritmanın ortalama hata değerinden $\alpha=0.05$ anlamlılık düzeyinde küçük olup olmadığı test edilmiştir.

Tablo 6.36 Eşleştirilmiş-t testi analiz sonuçlarını göstermektedir. Tabloda koyu işaretli değerler karşılaştırılan algoritmaların RMSE değerleri arasında istatistiksel olarak anlamlı bir farklılık olduğunu ifade etmektedir ($p<0.05$). Tablodan da görüldüğü gibi elde edilen hata değerleri açısından sadece IM algoritması ile SFS algoritması RMSE değerleri arasında istatistiksel olarak anlamlı bir farklılık vardır ve ele alınan ilk algoritmanın (IM) RMSE değeri, diğer algoritmanın (SFS) RMSE değerinden daha küçüktür.

Tablo 6.36 Eşleştirilmiş-t testi p değerleri

	IM/SS	IM/SFS	SS/SFS
p-değeri	0.103	0.012	0.182

Yeni nesil metasezgisel algoritmalar ayrıca Weka veri madenciliği yazılımında bulunan bazı regresyon tabanlı algoritmalar ve YSA modelleriyle karşılaştırılmıştır. Karşılaştırmaların adilane olması adına yeni nesil metasezgisellerde kullanılan aynı eğitim ve test kümeleri ve algoritma çalıştırma sayıları karşılaştırılan algoritmalarda da kullanılmıştır. Bu algoritmalar İzotonik Regresyon (IR), Gauss Süreçleri (GP), Tempo Regresyon (PR), Doğrusal Regresyon (LR), Radyal Tabanlı Fonksiyon Ağı (RBFN), En Düşük Medyan Karesi Doğrusal Regresyon (LeastMedSq), Destek Vektör Makinesi (SMOrg) ve Çok Katmanlı Algılayıcı (MLP)'dir. Tablo 6.37 karşılaştırma sonuçlarını göstermektedir. Tablodaki hata değerleri her bir algoritma için 30 çalıştırmanın ortalamasıdır. Tablodan görülebileceği gibi 11 algoritma içinde en düşük RMSE değerini IM algoritması vermiştir. SFS algoritmasının elde ettiği değer IM algoritmasına oldukça yakındır. Üçüncü sırada ise SS algoritması yer almaktadır. Karşılaştırmada dikkate alınan regresyon tabanlı algoritmalar ve YSA modelleri yeni nesil metasezgisel algoritmalarından daha yüksek hata değerleri vermiştir.

Tablo 6.37 Karşılaştırma Sonuçları

Algoritma	IR	GP	PR	LR	RBFN	LeastMedSq
RMSE	18465.97	18494.45	18509.97	18509.97	18542.66	18620.27
Algoritma	SMOrg	MLP	IM	SS	SFS	
RMSE	18700.44	21311.99	18107.73	18252.86	18154.37	

Tablo 6.37'de ele alınan algoritmaların elde ettiği ortalama hata değerleri arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını test etmek için Eşleştirmeli-t testi yapılmıştır. Tablo 6.38 Eşleştirmeli-t testi analiz sonuçlarını göstermektedir. Tabloda koyu işaretli değerler karşılaştırılan algoritmaların RMSE değerleri arasında istatistiksel olarak anlamlı bir farklılık olduğunu ifade etmektedir ($p < 0.05$).

Tablo 6.38 RMS değerleri için Eşleştirmeli-t testi p değerleri

Algoritmalar	IM/IR	IM/GP	IM/PR	IM/LR	IM/RBFN
p değeri	0.000	0.000	0.000	0.000	0.000
Algoritmalar	IM/LeastMedSq	IM/SMOrg	IM/MLP	IM/SS	IM/SFS
p değeri	0.000	0.000	0.000	0.103	0.012
Algoritmalar	SS/IR	SS/GP	SS/PR	SS/LR	SS/RBFN
p değeri	0.029	0.016	0.012	0.012	0.006
Algoritmalar	SS/LeastMedSq	SS/SMOrg	SS/MLP	SS/IM	SS/SFS
p değeri	0.001	0.000	0.000	0.897	0.818
Algoritmalar	SFS/IR	SFS/GP	SFS/PR	SFS/LR	SFS/RBFN
p değeri	0.000	0.000	0.000	0.000	0.000
Algoritmalar	SFS/LeastMedSq	SFS/SMOrg	SFS/MLP	SFS/IM	SFS/SS
p değeri	0.000	0.000	0.000	0.988	0.182

Tablo 6.38 incelendiğinde IM, SS ve SFS algoritmalarının Weka veri madenciliği yazılımındaki algoritmalarından daha iyi sonuçlar verdiği, p değerlerinin 0.05'ten daha küçük olmasından anlaşılabilmektedir. Bunun dışında IM algoritmasının SFS algoritmasından daha iyi sonuçlar verdiği de tablodan çıkarılabilmektedir. IM ile SS ve SS ile SFS arasında ise anlamlı bir fark yoktur.

Sonuç olarak gerçek hayat veri kümesinde testler sonucu en iyi performansı veren algoritma IM olmuştur. IM algoritmasına ilaveten bu çalışmada test edilen SS ve SFS algoritmaları da Weka veri madenciliği yazılımındaki algoritmalarından daha iyi sonuçlar vermiştir. Bu üç yeni nesil metasezgisel algoritmanın da diğerlerinden farkı $p=0.05$ değerine göre anlamlıdır.

7. BÖLÜM

SONUÇLAR

Bu çalışmada veri madenciliği sınıflandırma alanında sık kullanılan bir yöntem olan yapay sinir ağlarının eğitimi üç yeni nesil metasezgisel algoritma kullanılarak ele alınmıştır. Daha önceleri geleneksel yöntemlerle eğitilen YSA, daha sonraları metasezgisel algoritmalarla da eğitime başlanmıştır. Literatürde YSA eğitimi için kullanılan geleneksel ve metasezgisel algoritmaların yanına bu çalışmayla yeni nesil metasezgisel algoritmalar da eklenmiştir ve performansları karşılaştırılmıştır. Böylece çalışmada ele alınan İyon hareketi algoritması, sosyal örümcek algoritması ve stokastik fraktal arama algoritması literatürde ilk kez yapay sinir ağlarının eğitimi için kullanılmıştır.

Bu algoritmalarından İyon hareketi algoritması ele alınan diğer yeni nesil metasezgisel algoritmalarından gerek literatür sınıflandırma problemleri üzerinde gerekse gerçek hayat fonksiyon tahminleme probleminde daha iyi sonuçlar vermiştir. Literatür sınıflandırma problemlerinde geleneksel YSA eğitim yöntemleri ve metasezgisel algoritmaları ile karşılaştırmalarda IM algoritması büyük üstünlük göstermiştir. Diğer yeni nesil metasezgisel algoritmalar ise karşılaştırılan algoritmalar ile rekabet edebilecek düzeyde sonuçlar elde etmiştir. Gerçek hayat fonksiyon tahminleme veri kümesinde ise ele alınan üç yeni nesil metasezgisel algoritmanın regresyon tabanlı modeller ve YSA modelleriyle karşılaştırılmasında üç algoritmada karşılaştırılan algoritmalara tahminleme yönünden baskınlık sağlamıştır.

Yapılan deneysel çalışmalar ve istatistiksel analizler sonucunda ele alınan İyon hareketi algoritması, sosyal örümcek algoritması ve stokastik fraktal arama algoritmasının yapay sinir ağlarının eğitiminde başarılı bir şekilde kullanılabileceği ortaya konmuştur. Bu üç algoritma arasında IM algoritması bu haliyle YSA eğitiminde oldukça etkili bir

algoritmadır. YSA eğitimi dışında farklı alanlara da uygulanabilir ve iyi sonuçlar elde edilebilir. Ayrıca algoritmada bazı değişiklikler yapılarak veya başka algoritmalarla hibritlenerek daha da etkin bir hale getirilebilir.

Yapılacak gelecek çalışmalarda IM algoritmasının hibritleştirilerek başarı düzeyinin daha da artırılması ve farklı problemlere de uyarlanması hedeflenmektedir.

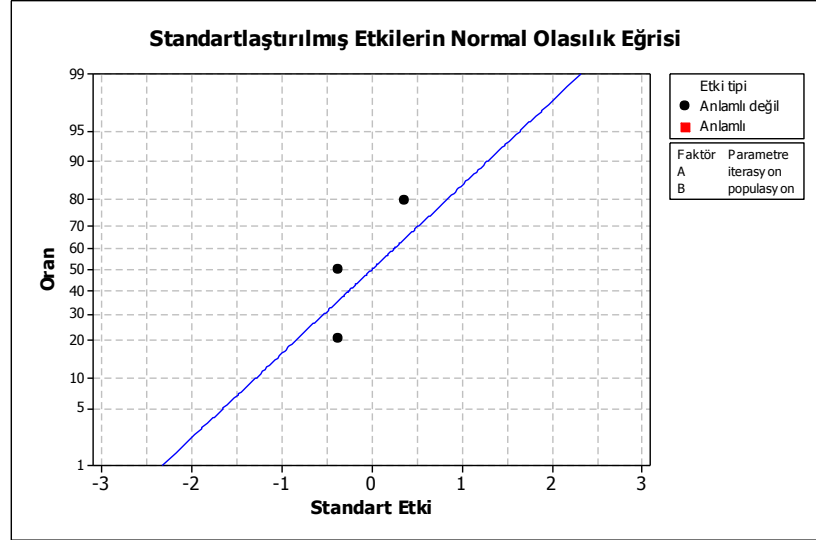
KAYNAKÇA

- [1] Ç. Elmas, *Yapay Sinir Ağları*. Ankara: Seçkin Yayınları, 2003.
- [2] E. Öztemel, *Yapay Sinir Ağları*. Papatya Yayıncılık, 2003.
- [3] S. Kulluk, “Karıncalar koloni optimizasyonu ile yapay sinir ağlarından kural çıkarımı”, Erciyes Üniversitesi, Kayseri, 2009.
- [4] J. Han, M. Kamber, ve J. Pei, *Data Mining: Concepts and Techniques, Third Edition*, 3 edition. Haryana, India; Burlington, MA: Morgan Kaufmann, 2011.
- [5] Z. Çakmak, N. Uzgören, ve G. Keçek, “Kümeleme Analizi Teknikleriyle İllerin Kültürel Yapılarına Göre Sınıflandırılması ve Değişimlerinin İncelenmesi”, *DPÜ Sos. Bilim. Derg.*, sayı 12, ss. 15–36, 2005.
- [6] F. Gürbüz, L. Özbakır, ve H. Yapıcı, “Türkiye’de Bir Havayolu İşletmesine Ait Parça Söküm Raporlarına İlişkin Veri Madenciliği Uygulaması”, *Gazi Üniversitesi Mühendis.-Mimar. Fakültesi Derg.*, c. 24, sayı 1, 2009.
- [7] H. Yurtoğlu, “Yapay sinir ağları metodolojisi ile öngörü modellemesi: bazı makroekonomik değişkenler için Türkiye örneği”, DPT, Ankara, 2005.
- [8] M. Yücesoy, “Temizlik kağıtları sektöründe yapay sinir ağları ile talep tahmini”, İstanbul Teknik Üniversitesi, İstanbul, 2011.
- [9] P. Yiğit, “Yapay sinir ağları ve kredi taleplerinin değerlendirilmesi üzerine bir uygulama”, İstanbul Üniversitesi, İstanbul, 2011.
- [10] M. K. Çelik, “Bankaların Finansal Başarısızlıklarının Geleneksel ve Yeni Yöntemlerle Öngörüsü”, *Önetim Ve Ekon. Celal Bayar Üniversitesi İktis. Ve İdari Bilim. Fakültesi Derg.*, c. 17, sayı 2, ss. 129–143, 2010.
- [11] S. Atasoy, “Yapay sinir ağları ve sinirsel bulanık ağlar ile insan kaynaklarında performans yönetimi modellenmesi”, Yıldız Teknik Üniversitesi, İstanbul, 2012.
- [12] R. S. Cintra ve H. F. de C. Velho, “Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model: Conventional Observation”, *ResearchGate*, c. 2, Tem. 2014.
- [13] A. Qadir, W. Semke, ve J. Neubert, “Vision Based Neuro-Fuzzy Controller for a Two Axes Gimbal System with Small UAV”, *J. Intell. Robot. Syst.*, c. 74, sayı 3–4, ss. 1029–1047, Haz. 2014.
- [14] A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu, ve M. Ahmadi, “Efficient hardware implementation of the hyperbolic tangent sigmoid function”, *Proc. - IEEE Int. Symp. Circuits Syst.*, ss. 2117–2120, Haz. 2009.

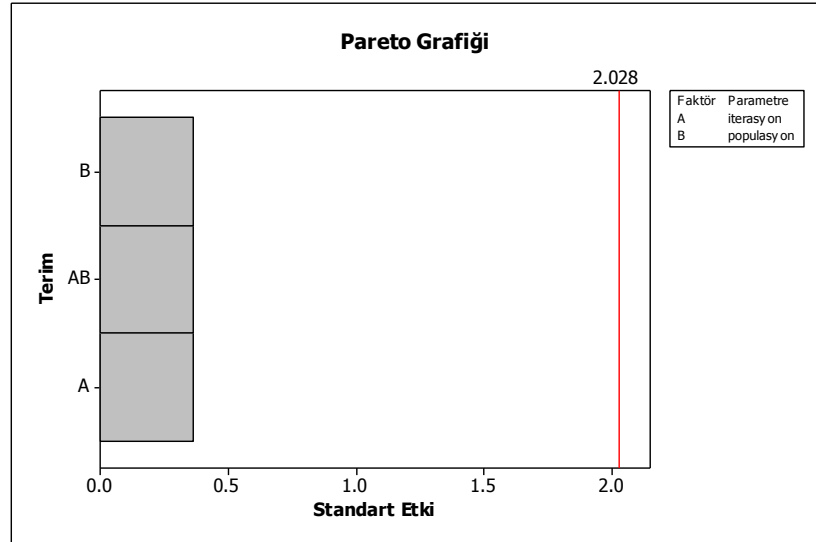
- [15] M. McCord-Nelson ve W. T. Illingworth, *A Practical Guide to Neural Nets*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1991.
- [16] M. T. Hagan, H. B. Demuth, M. H. Beale, ve O. D. Jesús, *Neural Network Design*, 2 edition. Place of publication not identified: Martin Hagan, 2014.
- [17] D. V. Negrov, I. M. Karandashev, V. V. Shakirov, Y. A. Matveyev, W. L. Dunin-Barkowski, ve A. V. Zenkevich, “A Plausible Memristor Implementation of Deep Learning Neural Networks”, Kas. 2015.
- [18] R. S. Sexton ve J. N. D. Gupta, “Comparative evaluation of genetic algorithm and backpropagation for training neural networks”, *Inf. Sci.*, c. 129, sayı 1–4, ss. 45–59, Kas. 2000.
- [19] V. G. Gudise ve G. K. Venayagamoorthy, “Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks”, içinde *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS '03*, 2003, ss. 110–117.
- [20] C. Blum ve K. Socha, “Training feed-forward neural networks with ant colony optimization: an application to pattern classification”, 2005, s. 6 pp.
- [21] D. Kim, H. Kim, ve D. Chung, “A Modified Genetic Algorithm for Fast Training Neural Networks”, içinde *Advances in Neural Networks – ISNN 2005*, J. Wang, X. Liao, ve Z. Yi, Ed. Springer Berlin Heidelberg, 2005, ss. 660–665.
- [22] B. Dengiz, C. Alabas-Uslu, ve O. Dengiz, “A Tabu Search Algorithm for the Training of Neural Networks”, *J. Oper. Res. Soc.*, c. 60, sayı 2, ss. 282–291, 2009.
- [23] A. Kattan, R. Abdullah, ve R. A. Salam, “Harmony Search Based Supervised Training of Artificial Neural Networks”, içinde *Modelling and Simulation 2010 International Conference on Intelligent Systems*, 2010, ss. 105–110.
- [24] M. Zamani ve A. Sadeghi, “A Variation of Particle Swarm Optimization for Training of Artificial Neural Networks”, içinde *Computational Intelligence and Modern Heuristics*, A.-D. Ali, Ed. InTech, 2010.
- [25] S. Kulluk, L. Ozbakir, ve A. Baykasoglu, “Training neural networks with harmony search algorithms for classification problems”, *Eng. Appl. Artif. Intell.*, c. 25, sayı 1, ss. 11–19, Şub. 2012.
- [26] R. C. Green II, L. Wang, ve M. Alam, “Training neural networks using Central Force Optimization and Particle Swarm Optimization: Insights and comparisons”, *Expert Syst. Appl.*, c. 39, sayı 1, ss. 555–563, Oca. 2012.
- [27] S. Nandy, P. P. Sarkar, ve A. Das, “Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training”, *ArXiv12065360 Cs*, Haz. 2012.

- [28] S. Nandy, P. P. Sarkar, ve A. Das, “Training a Feed-forward Neural Network with Artificial Bee Colony Based Backpropagation Method”, *Int. J. Comput. Sci. Inf. Technol.*, c. 4, sayı 4, ss. 33–46, Ağu. 2012.
- [29] A. Askarzadeh ve A. Rezazadeh, “Artificial neural network training using a new efficient optimization algorithm”, *Appl. Soft Comput.*, c. 13, sayı 2, ss. 1206–1213, Şub. 2013.
- [30] M. Yaghini, M. M. Khoshraftar, ve M. Fallahi, “A hybrid algorithm for artificial neural network training”, *Eng. Appl. Artif. Intell.*, c. 26, sayı 1, ss. 293–301, Oca. 2013.
- [31] X.-S. Yang, “Harmony Search as a Metaheuristic Algorithm”, içinde *Music-Inspired Harmony Search Algorithm*, Z. W. Geem, Ed. Springer Berlin Heidelberg, 2009, ss. 1–14.
- [32] B. Javidy, A. Hatamlou, ve S. Mirjalili, “Ions motion algorithm for solving optimization problems”, *Appl. Soft Comput.*, c. 32, ss. 72–79, Tem. 2015.
- [33] J. J. Q. Yu ve V. O. K. Li, “A social spider algorithm for global optimization”, *Appl. Soft Comput.*, c. 30, ss. 614–627, May. 2015.
- [34] H. Salimi, “Stochastic Fractal Search: A powerful metaheuristic algorithm”, *Knowl.-Based Syst.*, c. 75, ss. 1–18, Şub. 2015.
- [35] D. Karaboga ve C. Ozturk, “Neural networks training by Artificial Bee Colony algorithm on pattern classification”, *Neural Netw. World*, c. 19, sayı 3, ss. 279–292, Oca. 2009.
- [36] S. Kulluk, “A novel hybrid algorithm combining hunting search with harmony search algorithm for training neural networks”, *J. Oper. Res. Soc.*, c. 64, sayı 5, ss. 748–761, May. 2013.

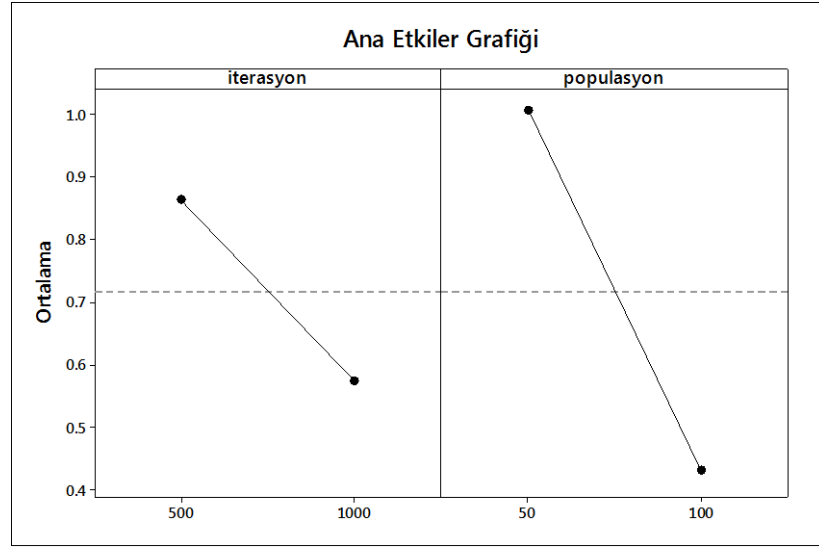
EKLER



Ek 1 IM algoritması Cancer veri kümesi normal olasılık eğrisi



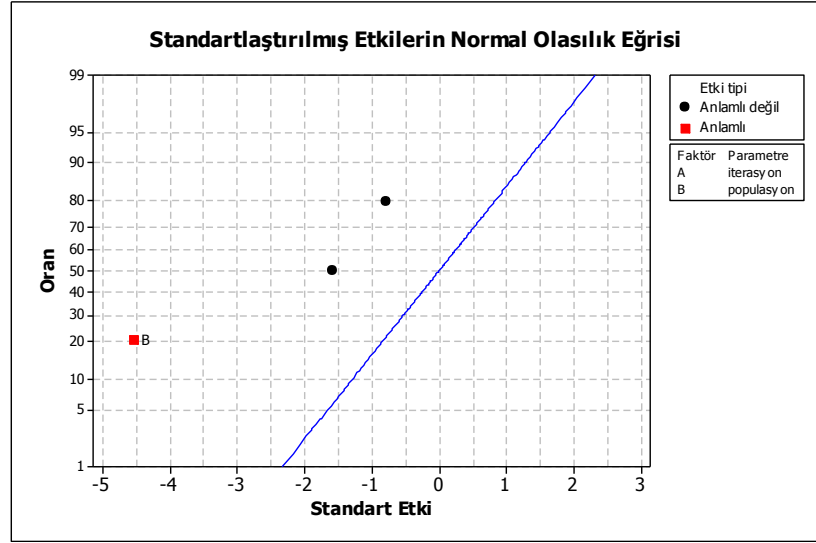
Ek 2 IM algoritması Cancer veri kümesi pareto grafiği



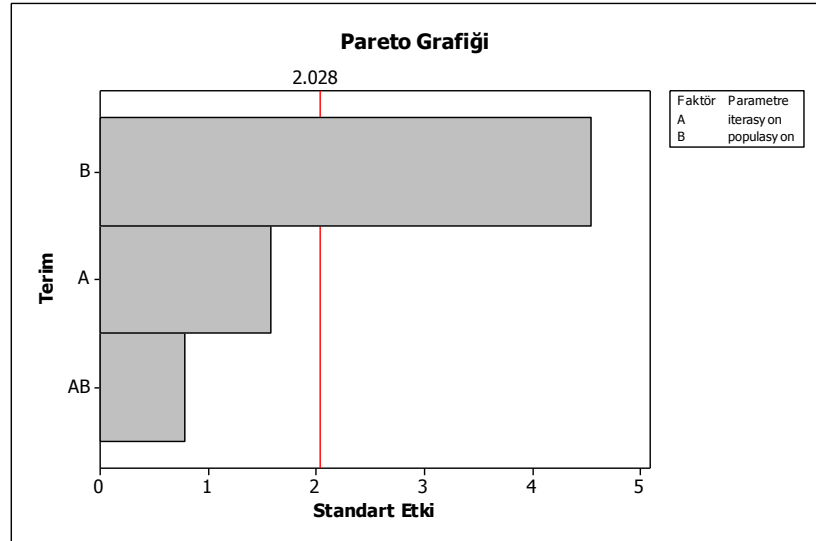
Ek 3 IM Algoritması Cancer veri kümesi tekli parametre etkileri



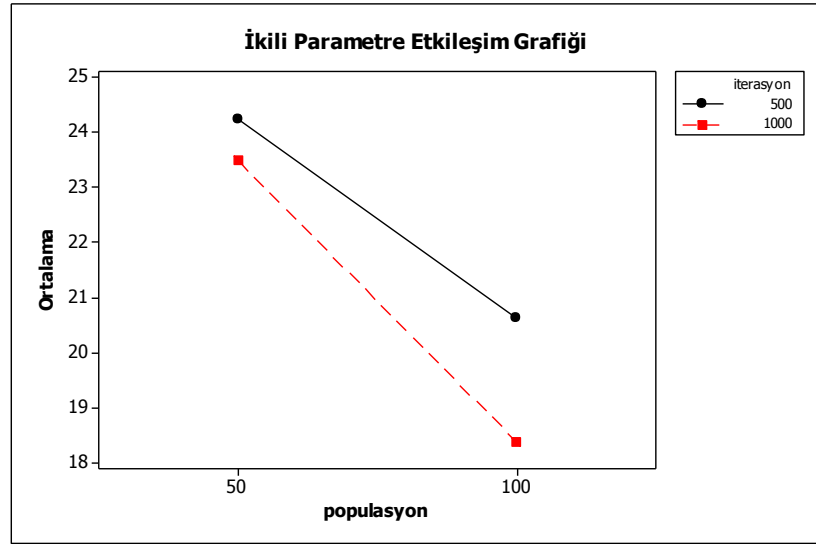
Ek 4 IM algoritması Cancer veri kümesi ikili parametre etkileşimleri



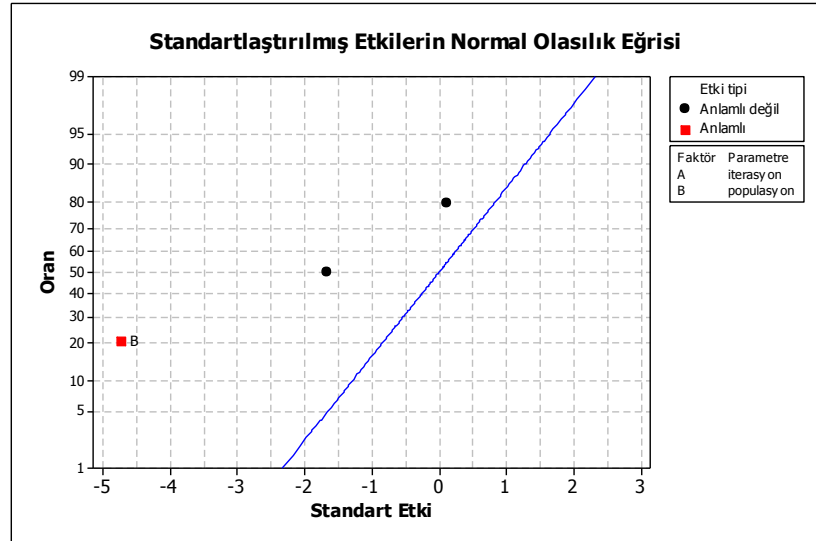
Ek 5 IM algoritması Card veri kümesi normal olasılık eğrisi



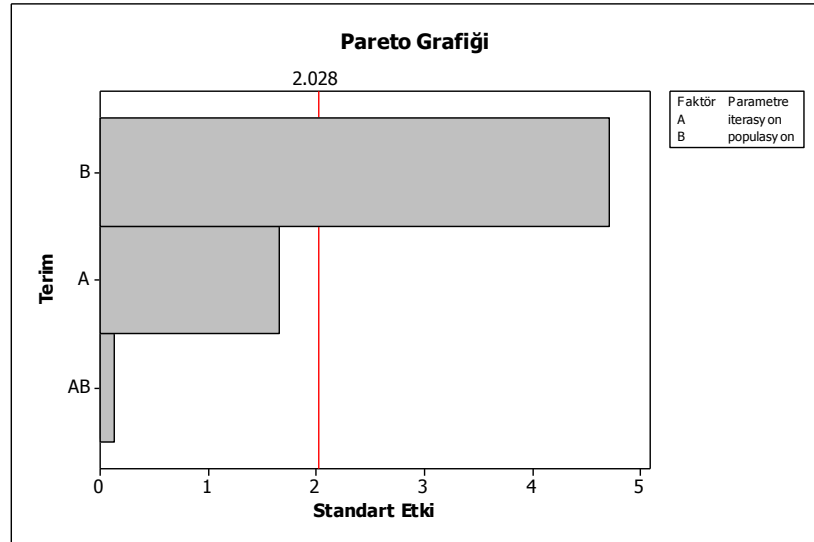
Ek 6 IM algoritması Card veri kümesi pareto grafiği



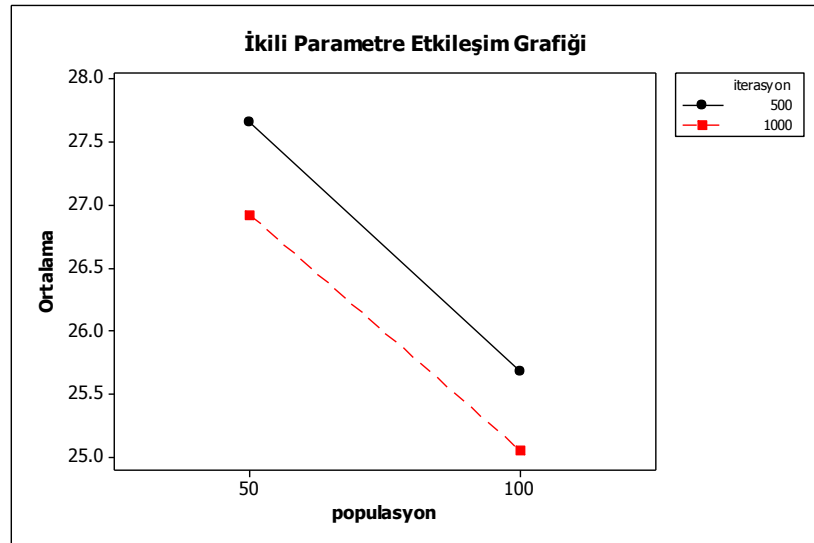
Ek 7 IM algoritması Card veri kümesi ikili parametre etkileşimleri



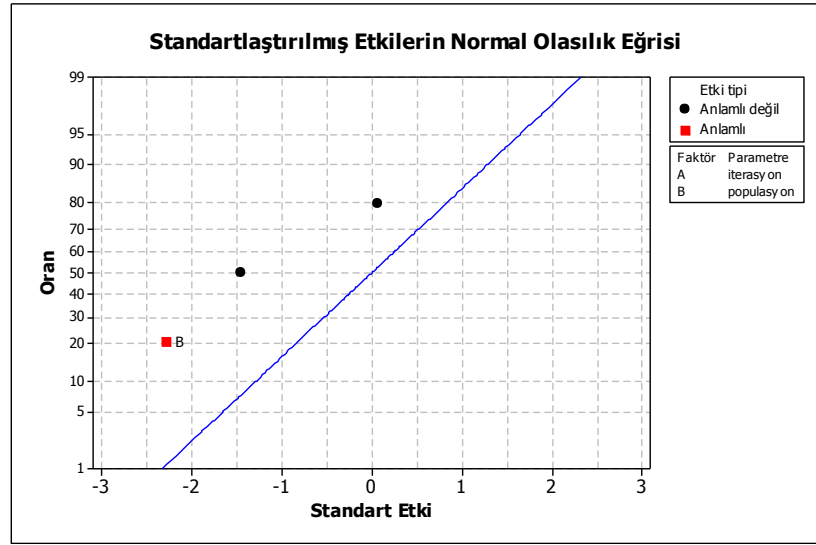
Ek 8 IM algoritması Diabetes veri kümesi normal olasılık eğrisi



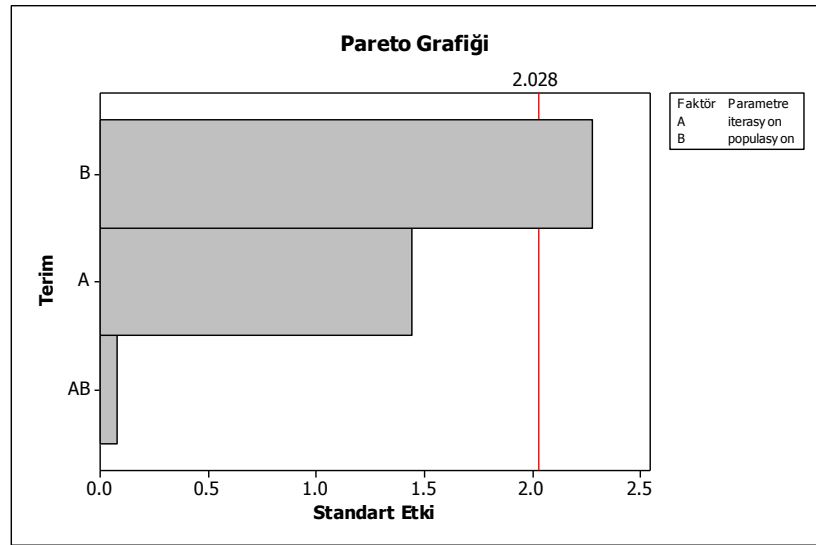
Ek 9 IM algoritması Diabetes veri kümesi pareto grafiği



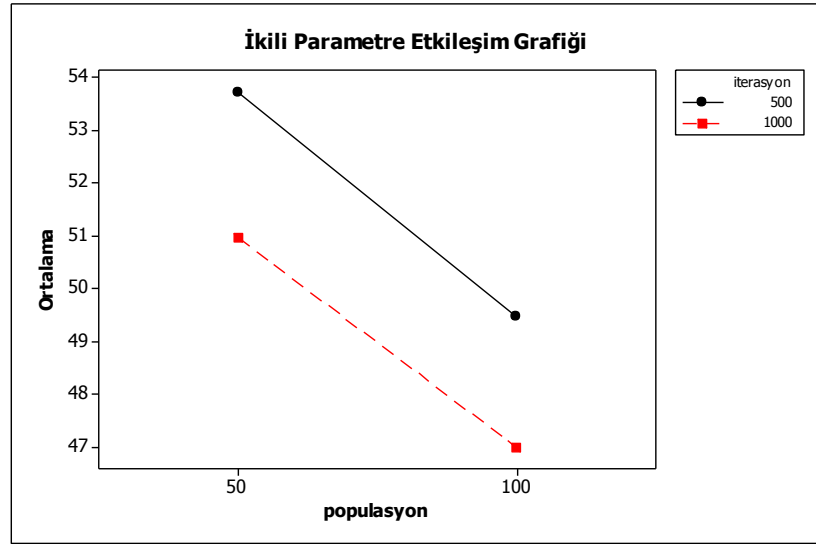
Ek 10 IM algoritması Diabetes veri kümesi ikili parametre etkileşimleri



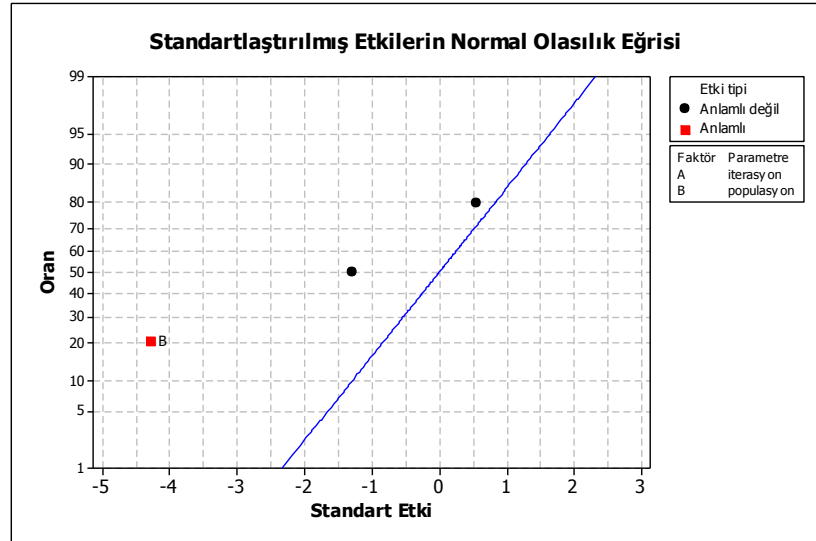
Ek 11 IM algoritması Glass veri kümesi normal olasılık eğrisi



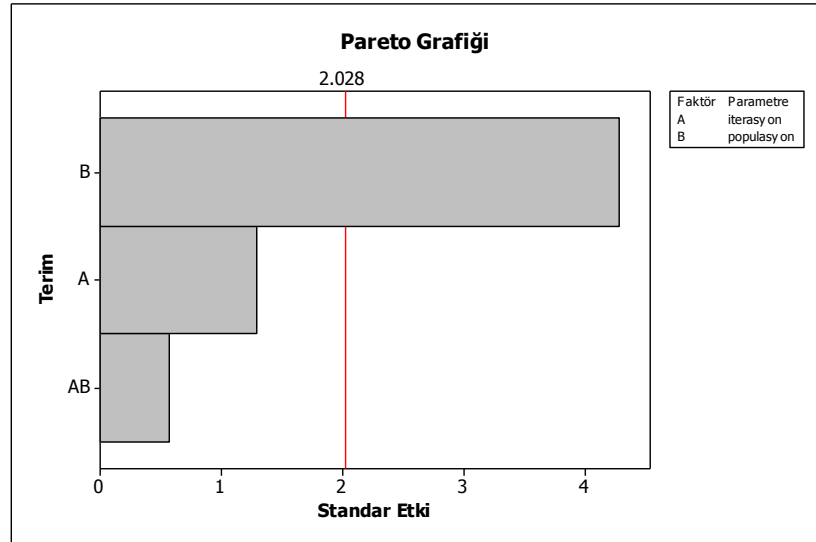
Ek 12 IM algoritması Glass veri kümesi pareto grafiği



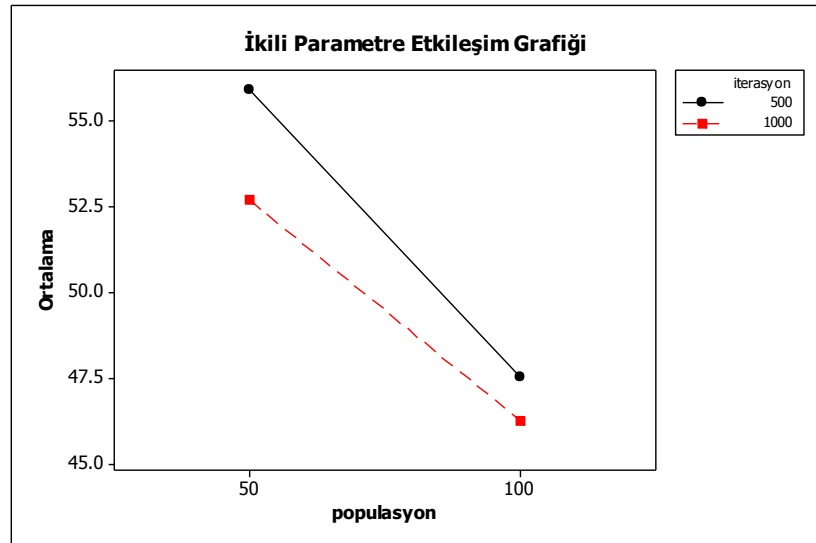
Ek 13 IM algoritması Glass veri kümesi ikili parametre etkileşimleri



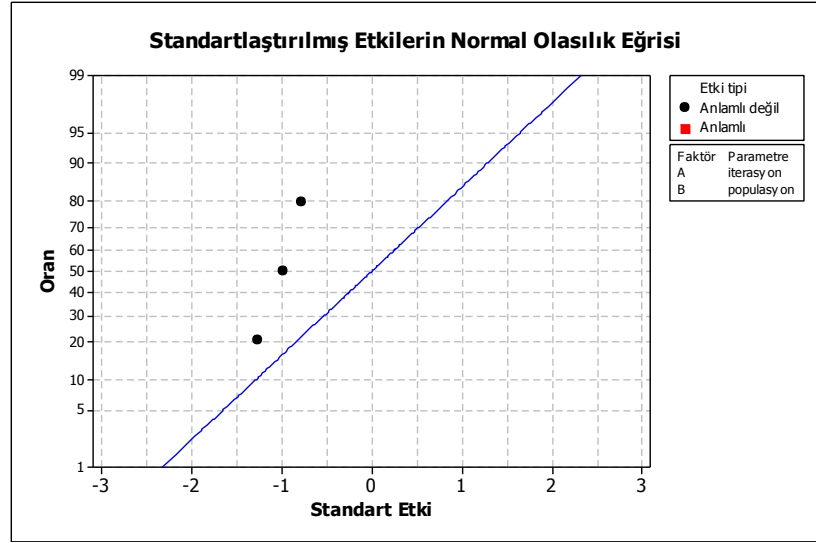
Ek 14 IM algoritması Soybean veri kümesi normal olasılık eğrisi



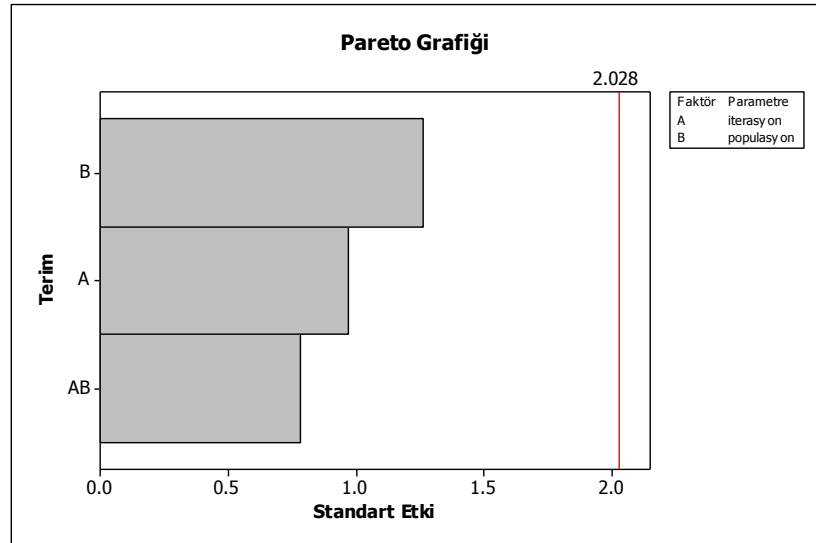
Ek 15 IM algoritması Soybean veri kümesi pareto grafiği



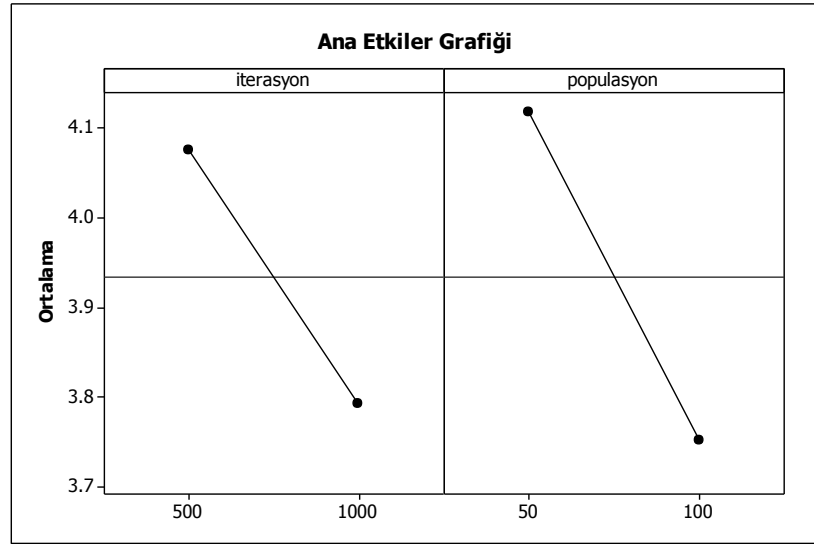
Ek 16 IM algoritması Soybean veri kümesi ikili parametre etkileşimleri



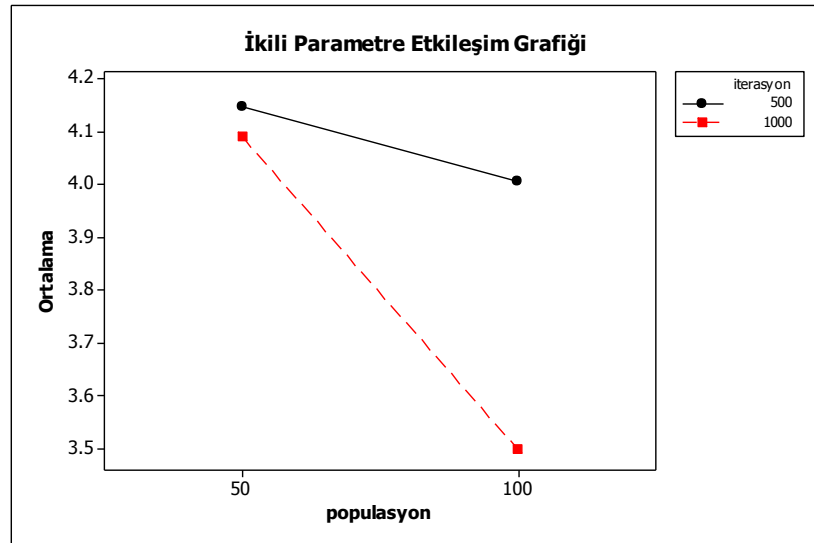
Ek 17 IM algoritması Thyroid veri kümesi normal olasılık eğrisi



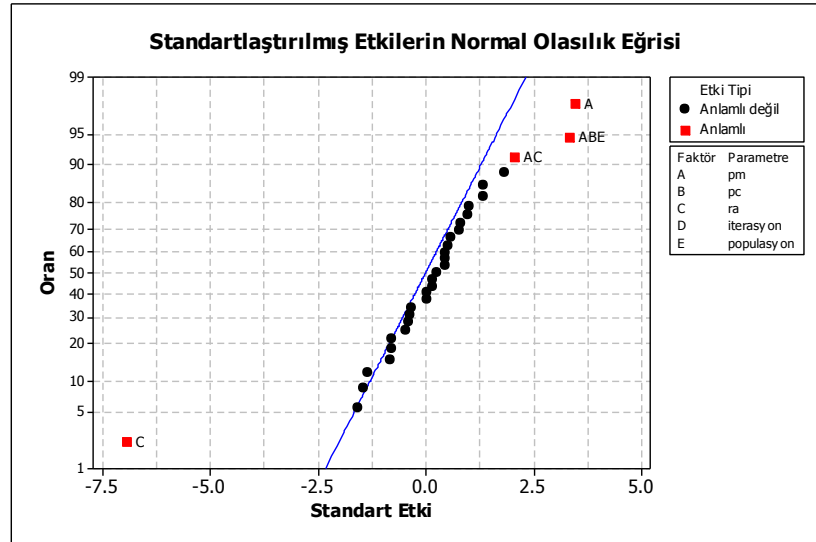
Ek 18 IM algoritması Thyroid veri kümesi pareto grafiği



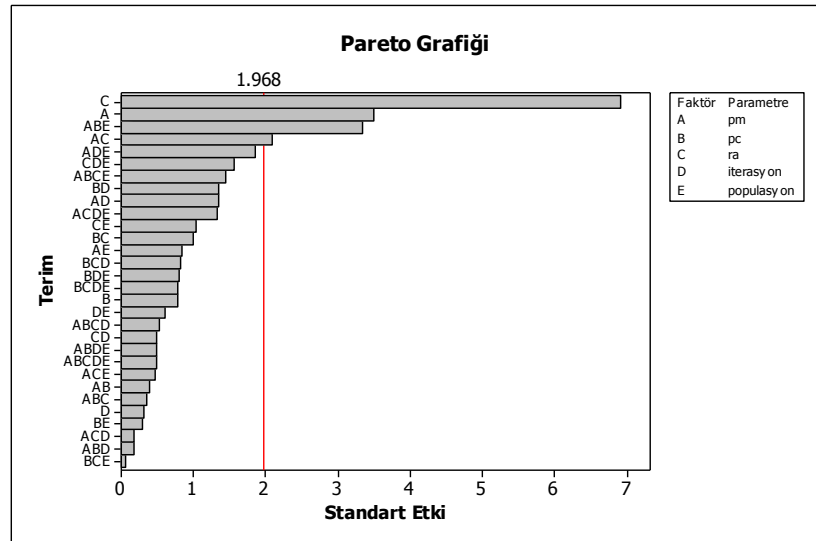
Ek 19 IM algoritması Thyroid veri kümesi tekli parametre etkileri



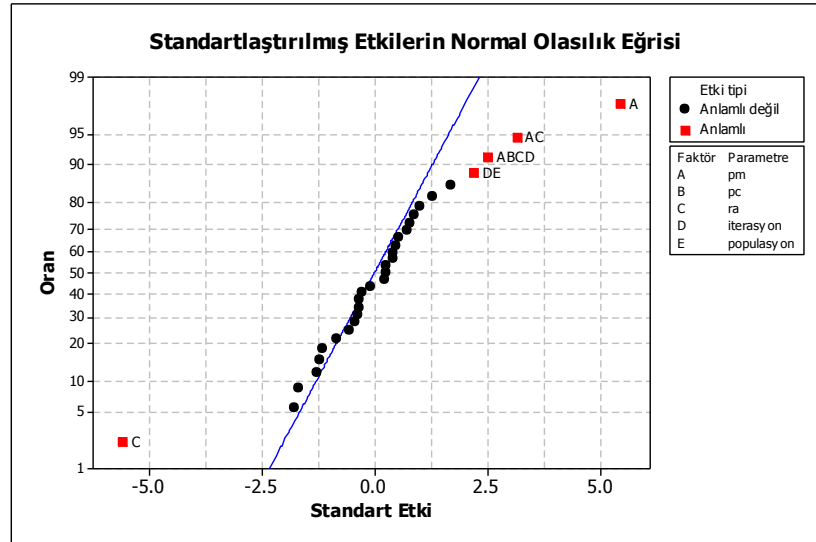
Ek 20 IM algoritması Thyroid veri kümesi ikili parametre etkileşimleri



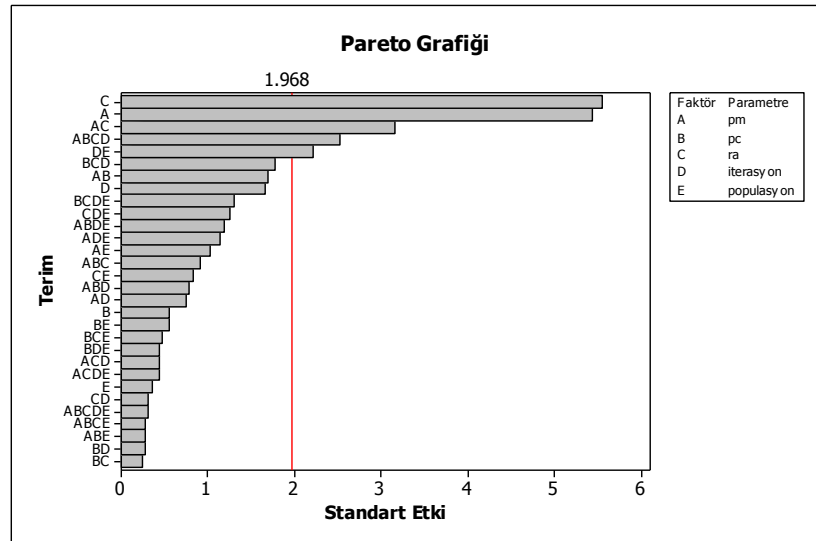
Ek 21 SS algoritması Card veri kümesi normal olasılık eğrisi



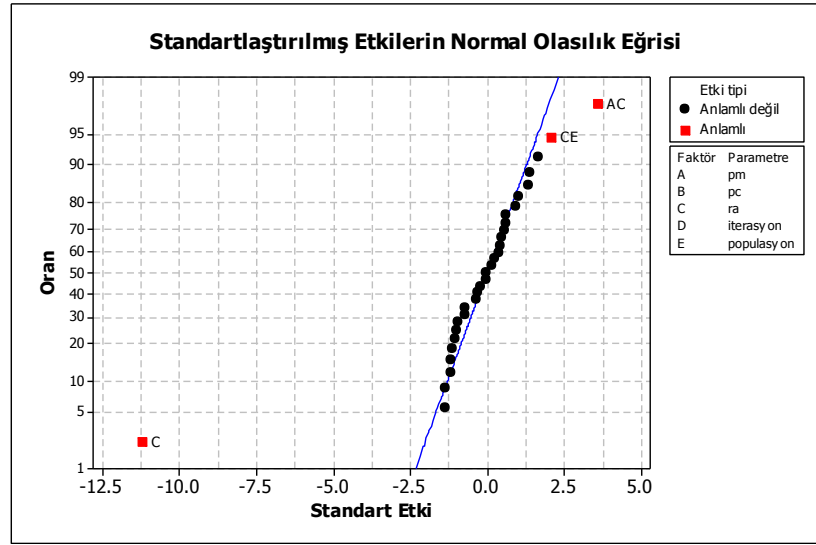
Ek 22 SS algoritması Card veri kümesi pareto grafiği



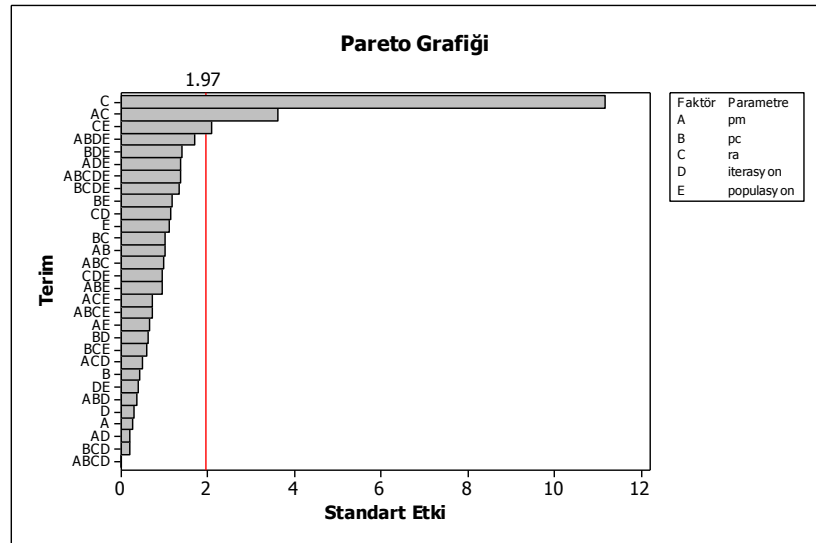
Ek 23 SS algoritması Diabetes veri kümesi normal olasılık eğrisi



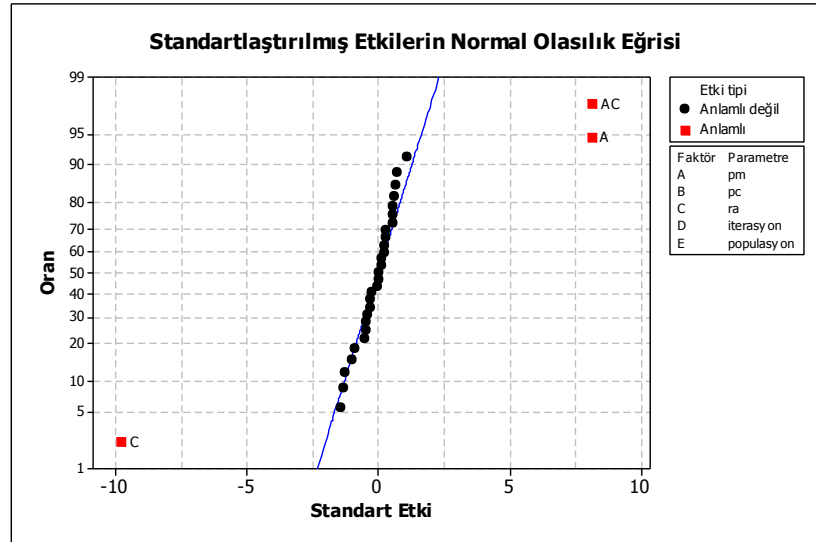
Ek 24 SS algoritması Diabetes veri kümesi pareto grafiği



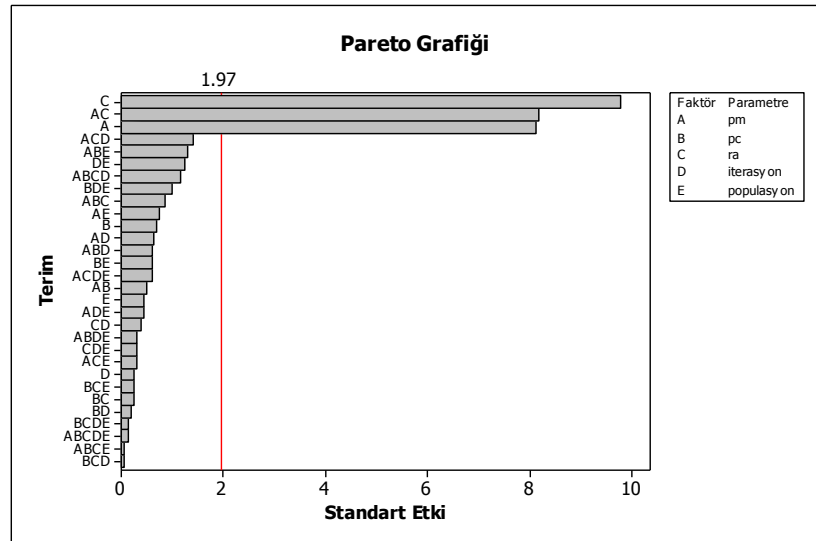
Ek 25 SS algoritması Gene veri kümesi normal olasılık dağılımı



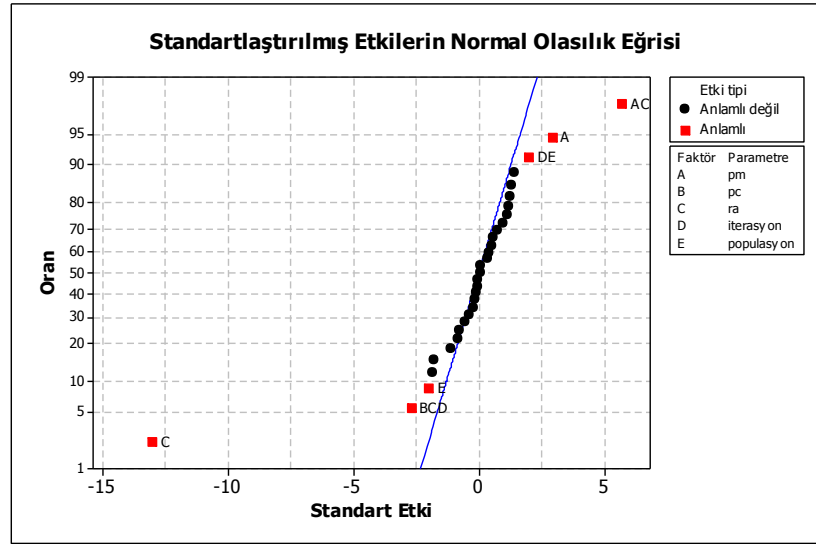
Ek 26 SS algoritması Gene veri kümesi pareto grafiği



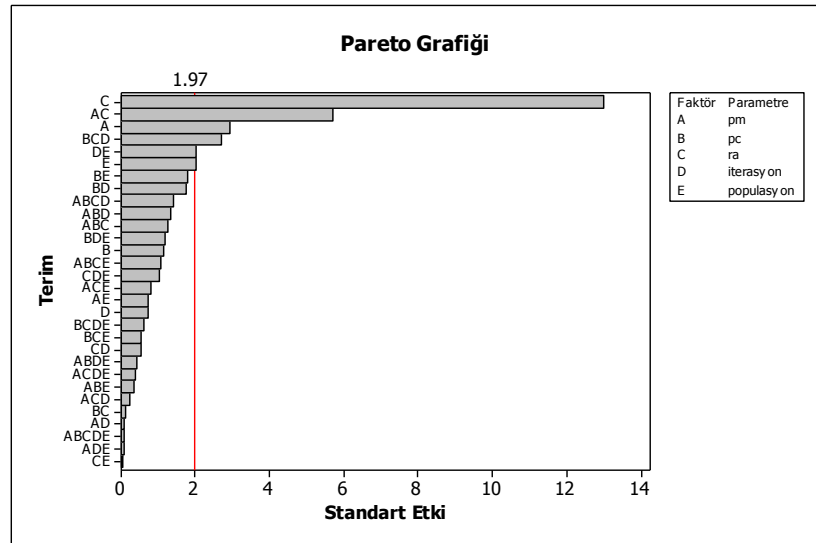
Ek 27 SS algoritması Glass veri kümesi normal olasılık eğrisi



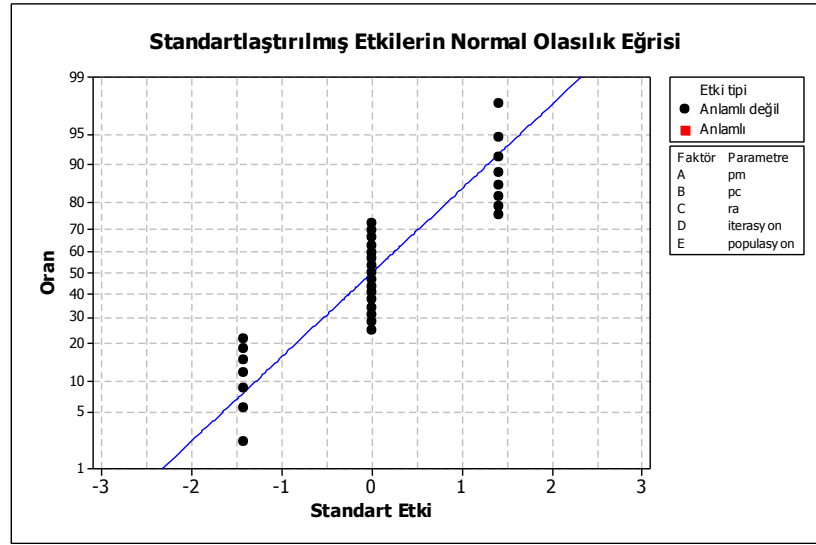
Ek 28 SS algoritması Glass veri kümesi pareto grafiği



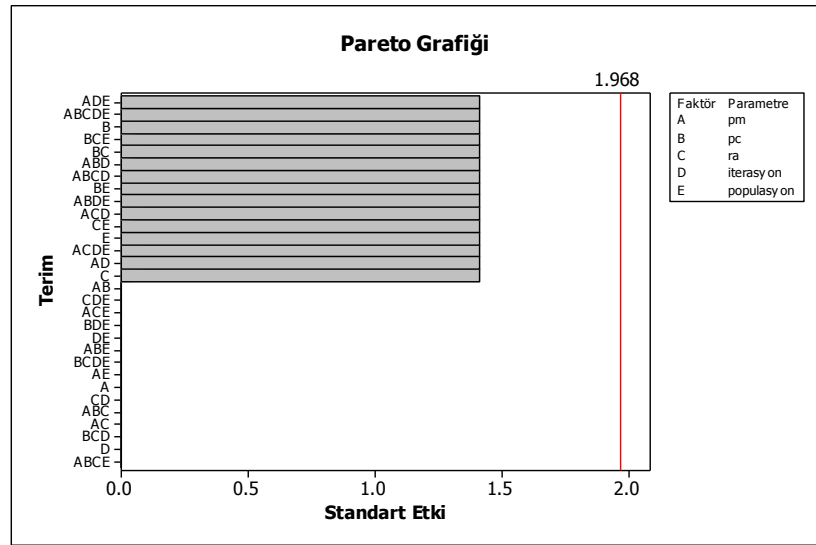
Ek 29 SS algoritması Soybean veri kümesi normal olasılık eğrisi



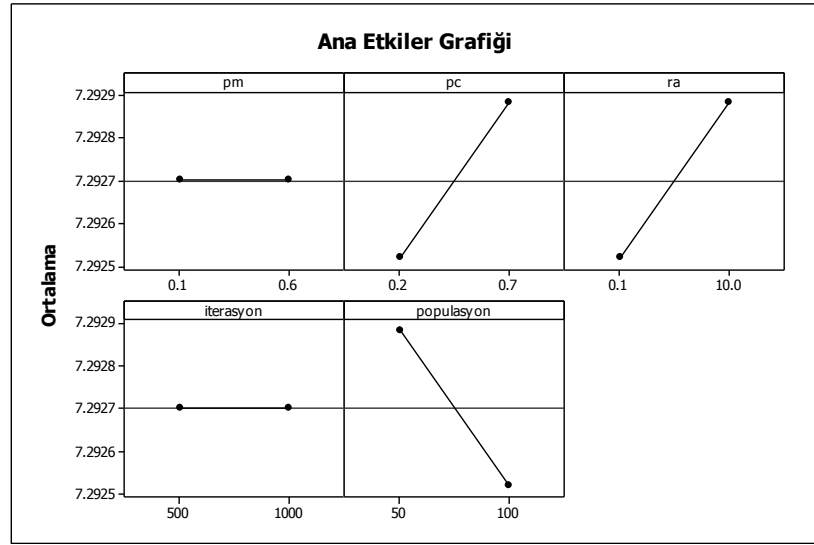
Ek 30 SS algoritması Soybean veri kümesi pareto grafiği



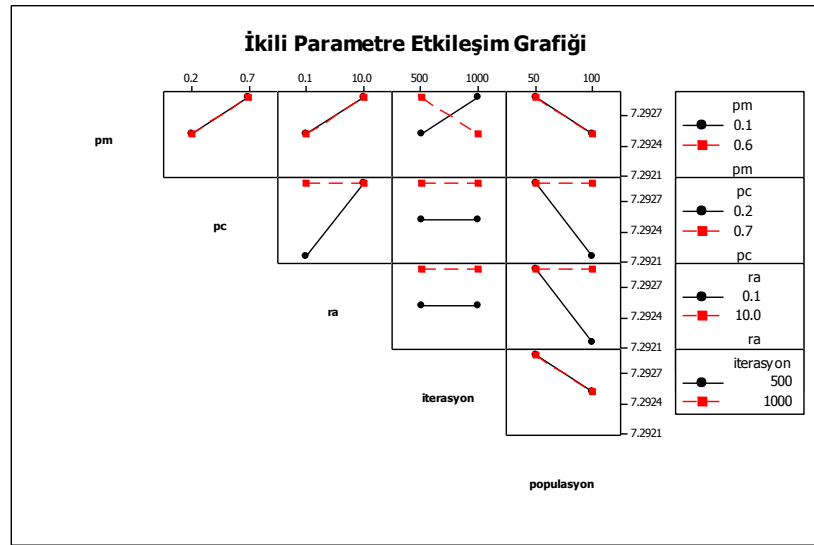
Ek 31 SS algoritması Thyroid veri kümesi normal olasılık eğrisi



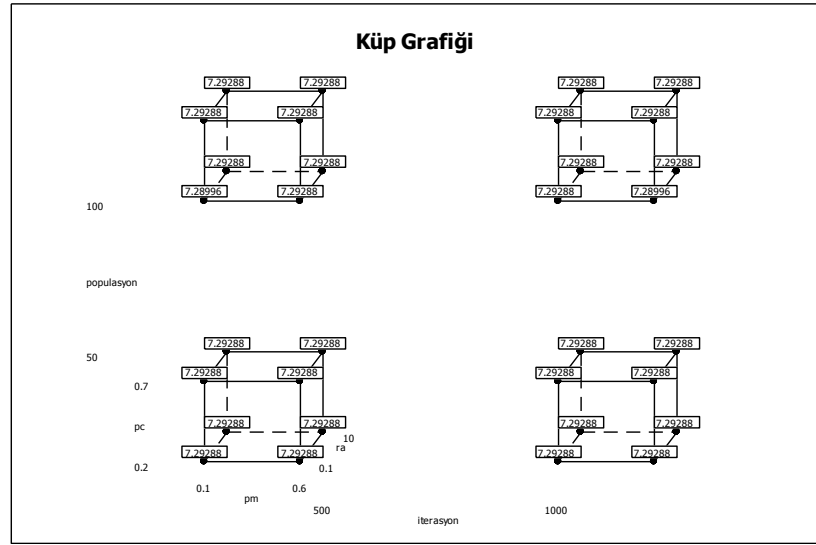
Ek 32 SS algoritması Thyroid veri kümesi pareto grafiği



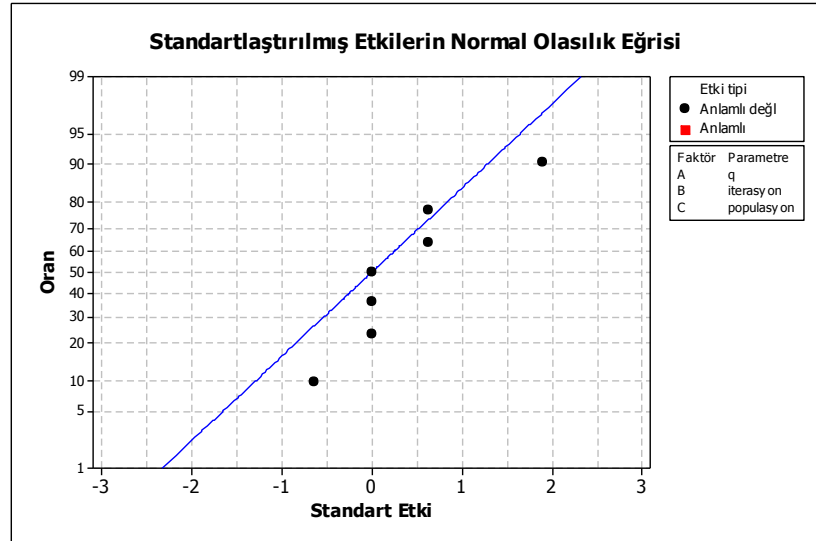
Ek 33 SS algoritması Thyroid veri kümesi tekli parametre etkileri



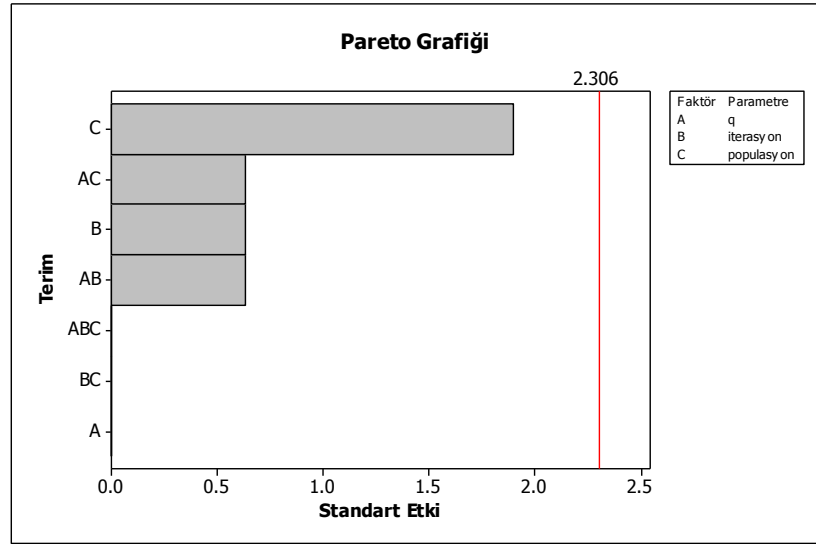
Ek 34 SS algoritması Thyroid veri kümesi ikili parametre etkileşimleri



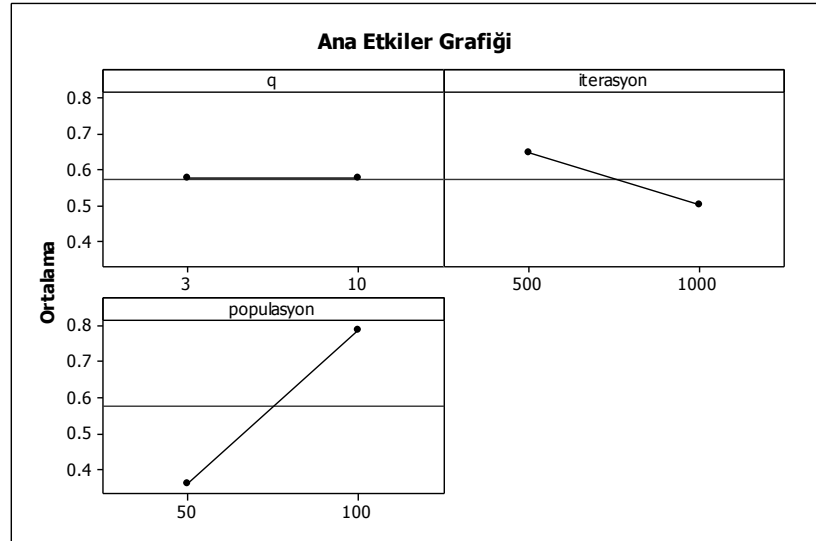
Ek 35 SS algoritması Thyroid veri kümesi küp grafiği



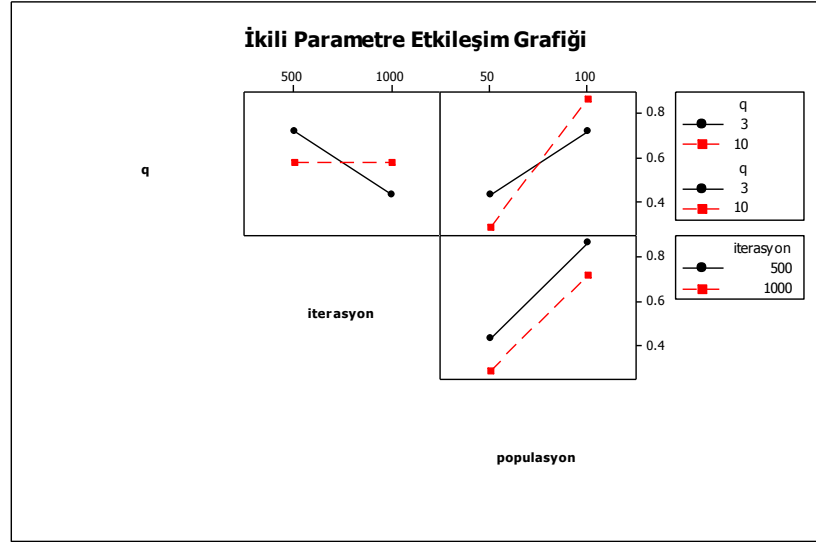
Ek 36 SFS algoritması Cancer veri kümesi normal olasılık eğrisi



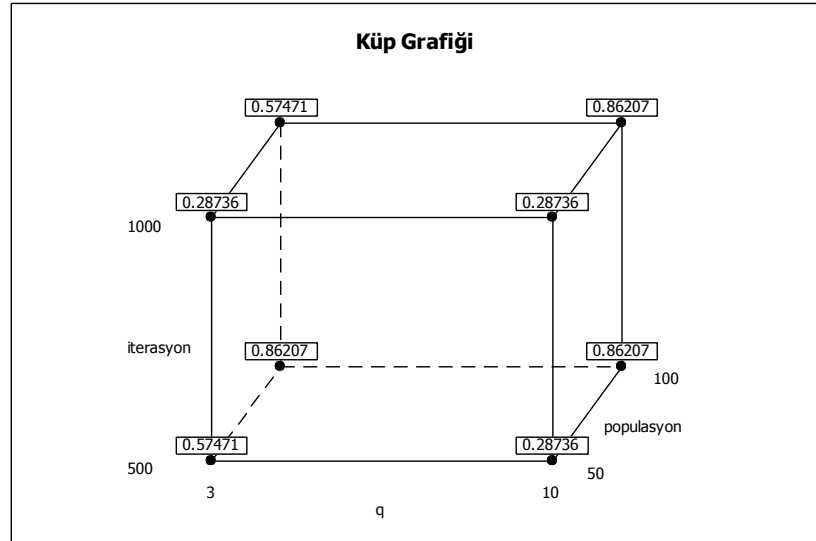
Ek 37 SFS algoritması Cancer veri kümesi pareto grafiği



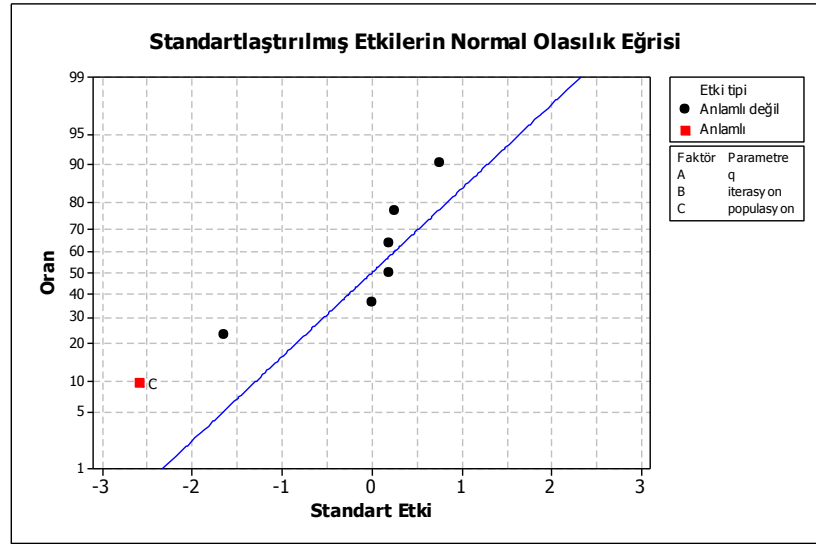
Ek 38 SFS algoritması Cancer veri kümesi tekli parametre etkileri



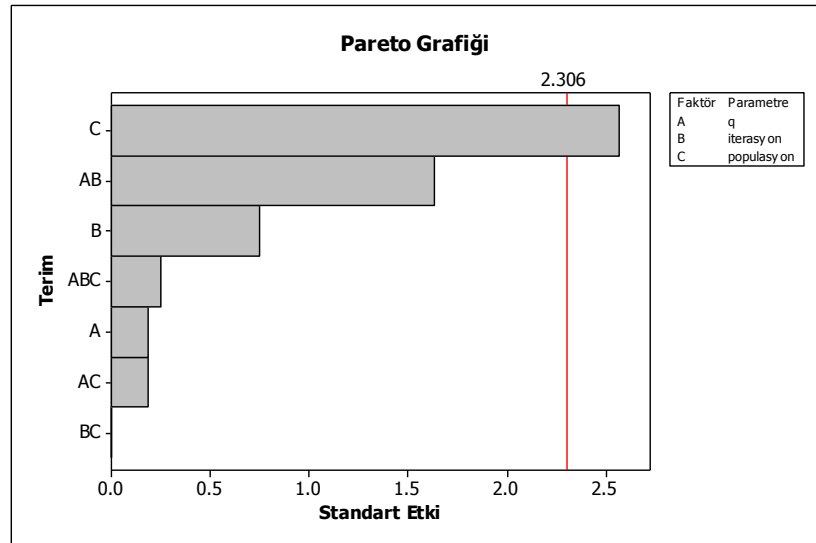
Ek 39 SFS algoritması Cancer veri kümesi ikili parametre etkileşimleri



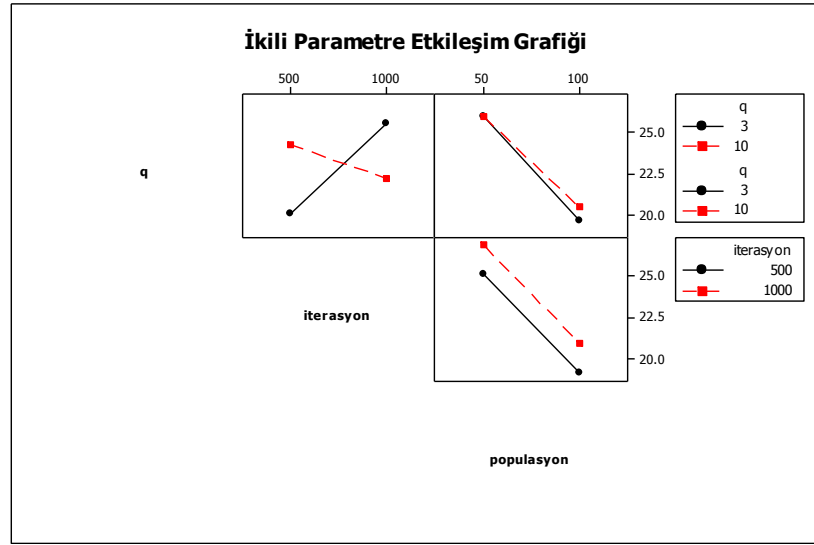
Ek 40 SFS algoritması Cancer veri kümesi küp grafiği



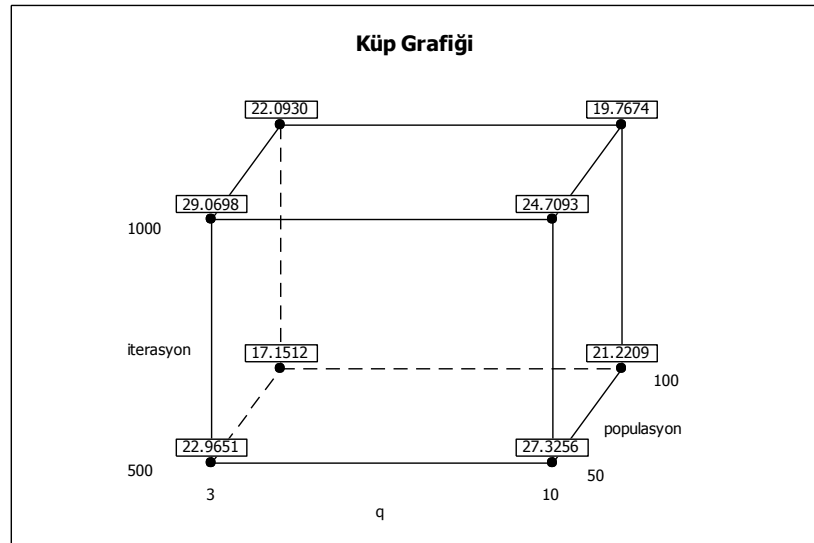
Ek 41 SFS algoritması Card veri kümesi normal olasılık eğrisi



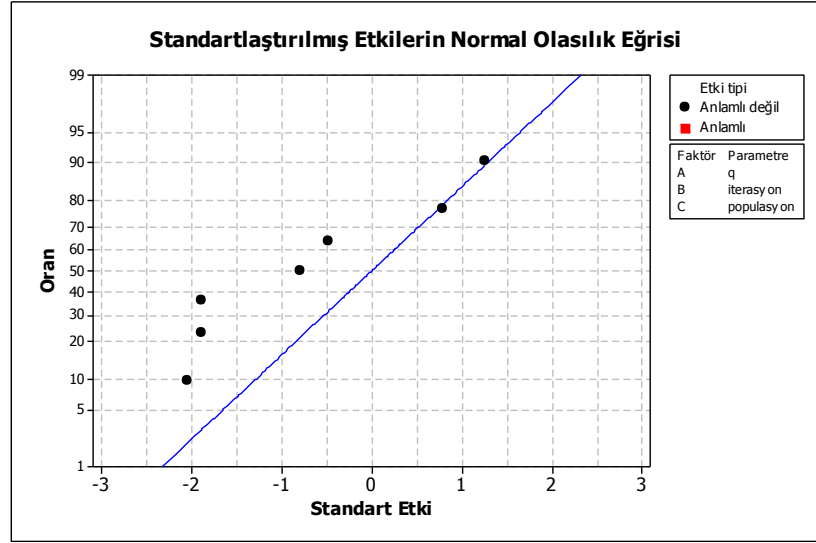
Ek 42 SFS algoritması Card veri kümesi pareto grafiği



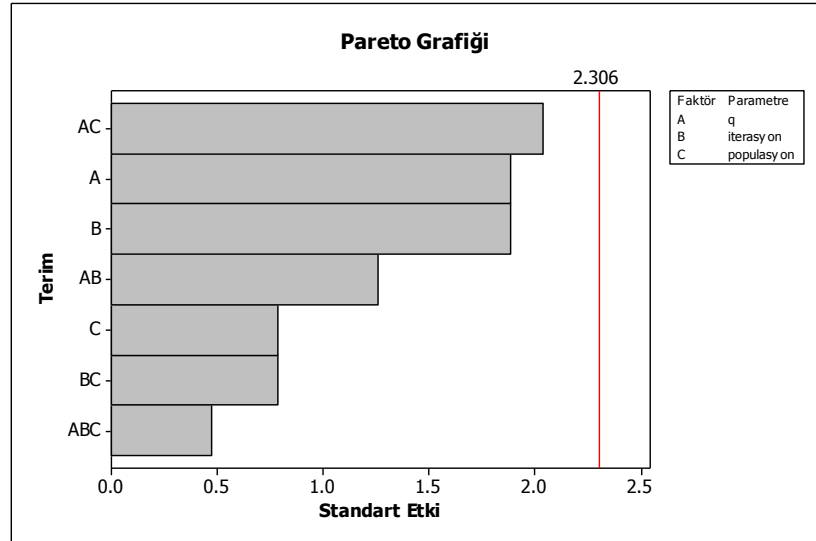
Ek 43 SFS algoritması Card veri kümesi ikili parametre etkileşimleri



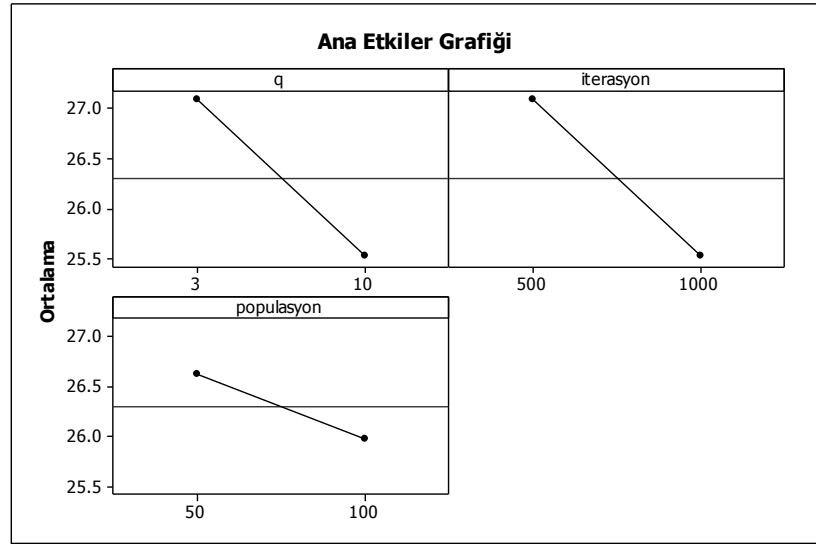
Ek 44 SFS algoritması Card veri kümesi küp grafiği



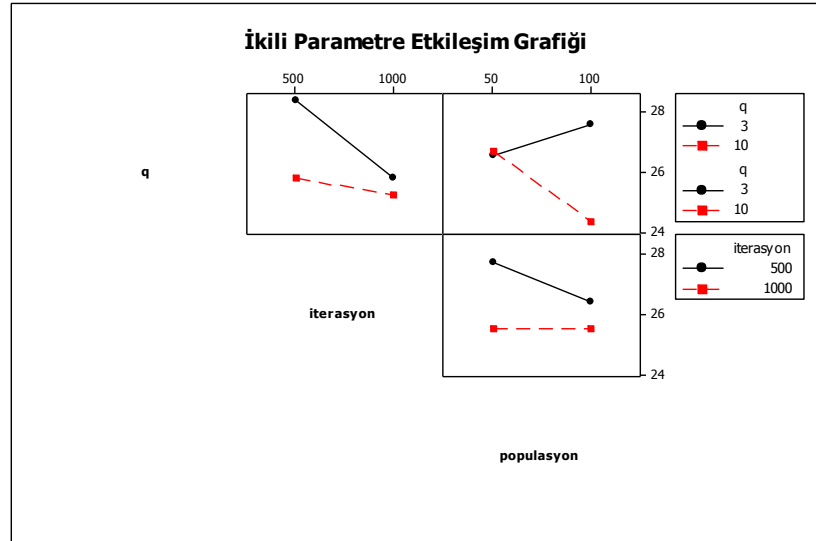
Ek 45 SFS algoritması Diabetes veri kümesi normal olasılık eğrisi



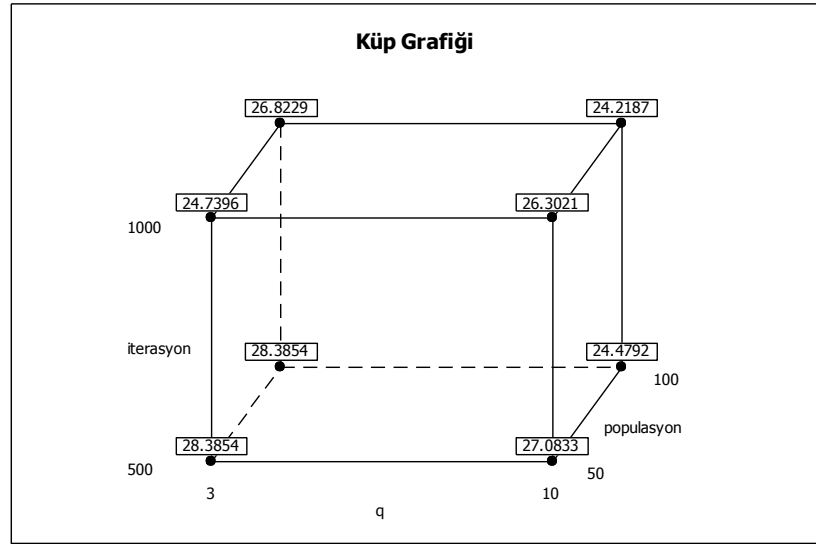
Ek 46 SFS algoritması Diabetes veri kümesi pareto grafiği



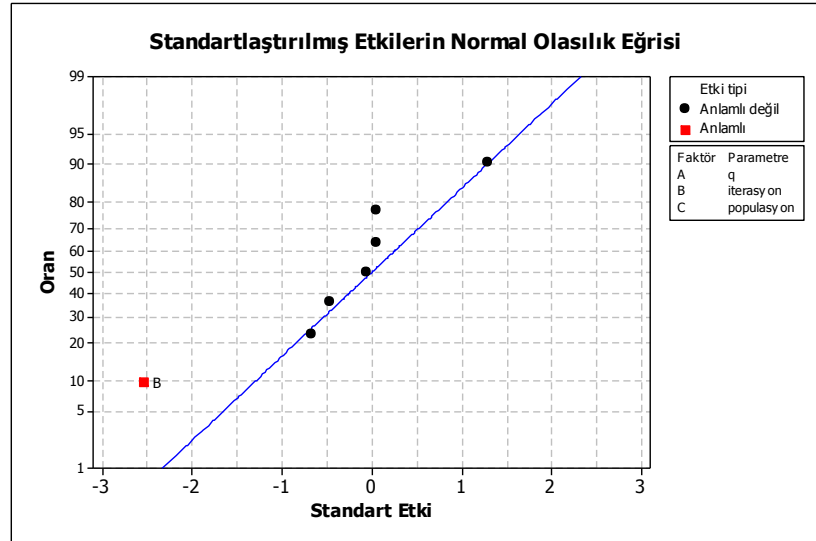
Ek 47 SFS algoritması Diabetes veri kümesi tekli parametre etkileri



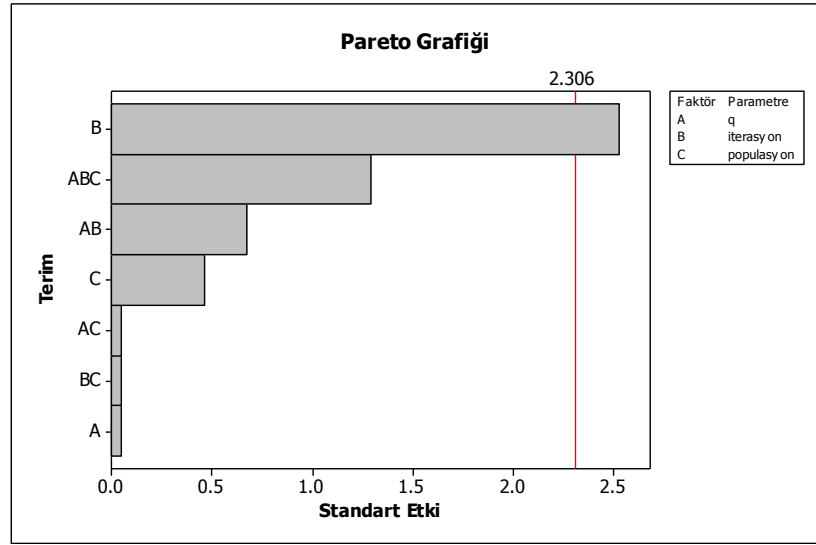
Ek 48 SFS algoritması Diabetes veri kümesi ikili parametre etkileşimleri



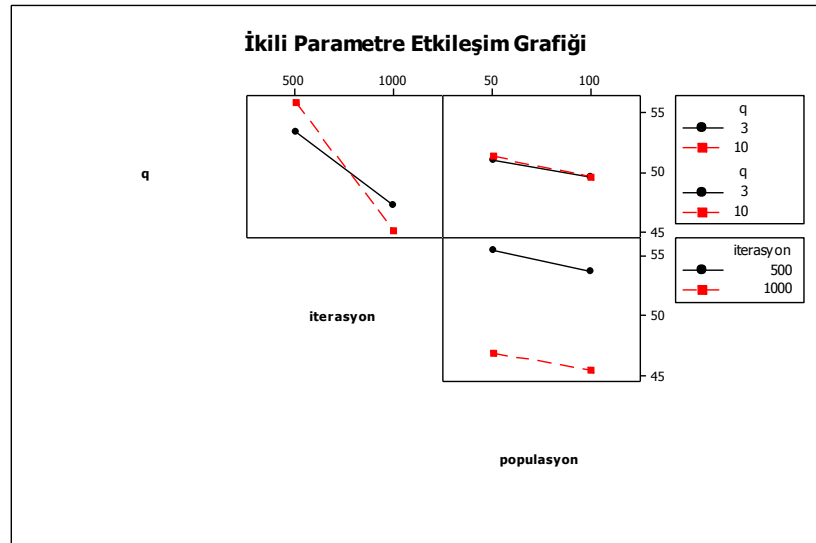
Ek 49 SFS algoritması Diabetes veri kümesi küp grafiği



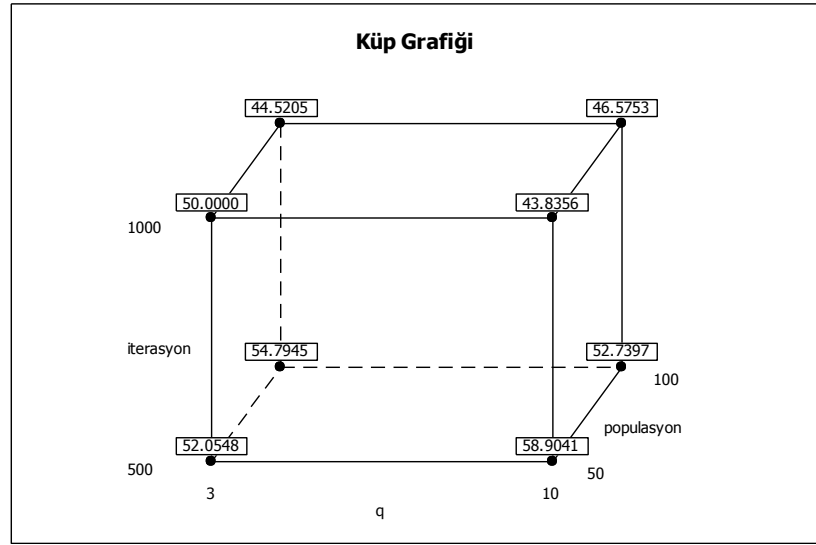
Ek 50 SFS algoritması Glass veri kümesi normal olasılık eğrisi



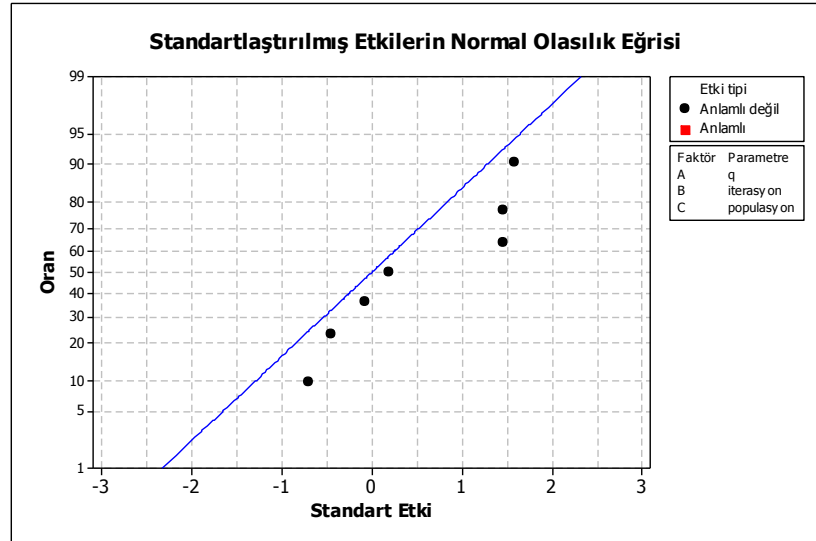
Ek 51 SFS algoritması Glass veri kümesi pareto grafiği



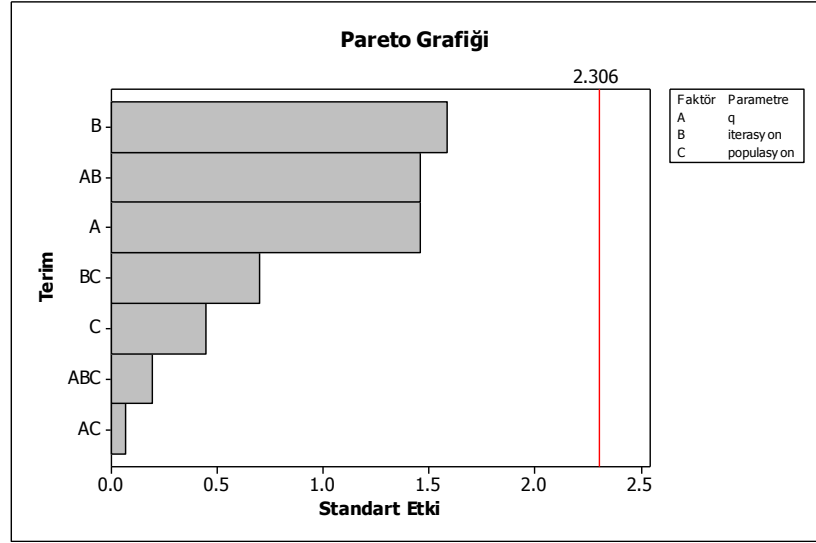
Ek 52 SFS algoritması Glass veri kümesi ikili parametre etkileşimleri



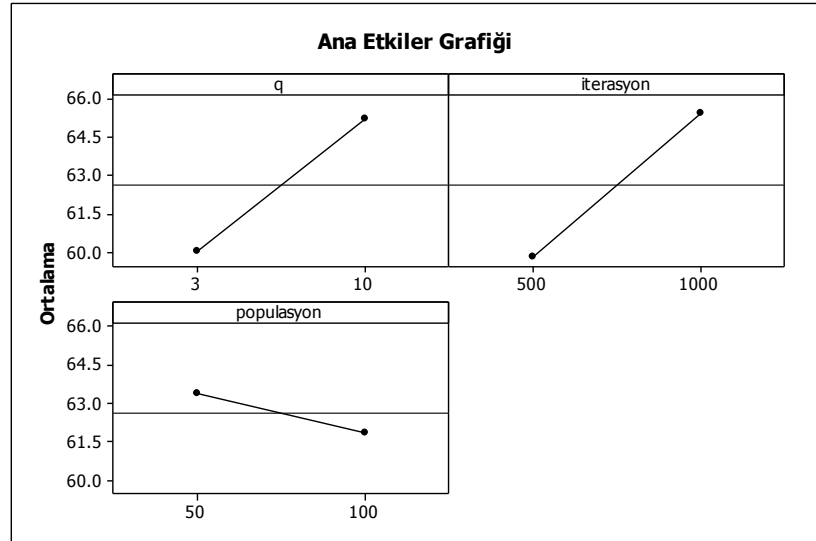
Ek 53 SFS algoritması Glass veri kümesi küp grafiği



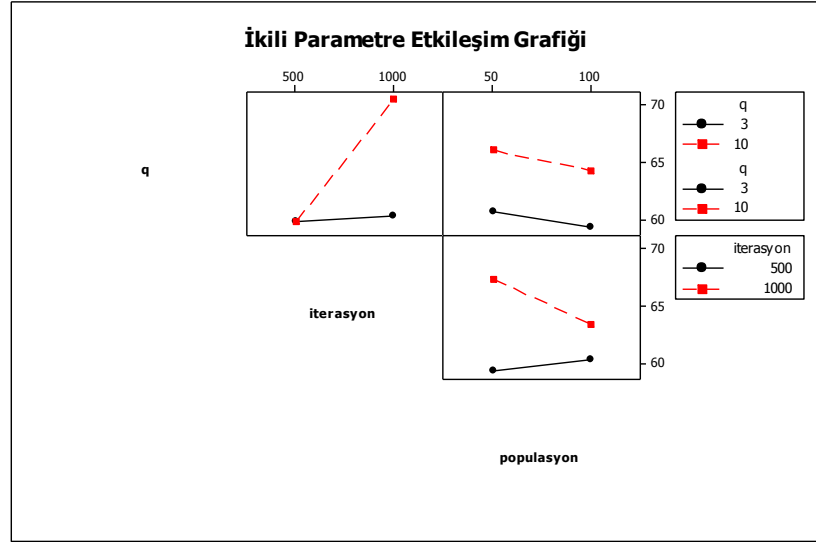
Ek 54 SFS algoritması Soybean veri kümesi normal olasılık eğrisi



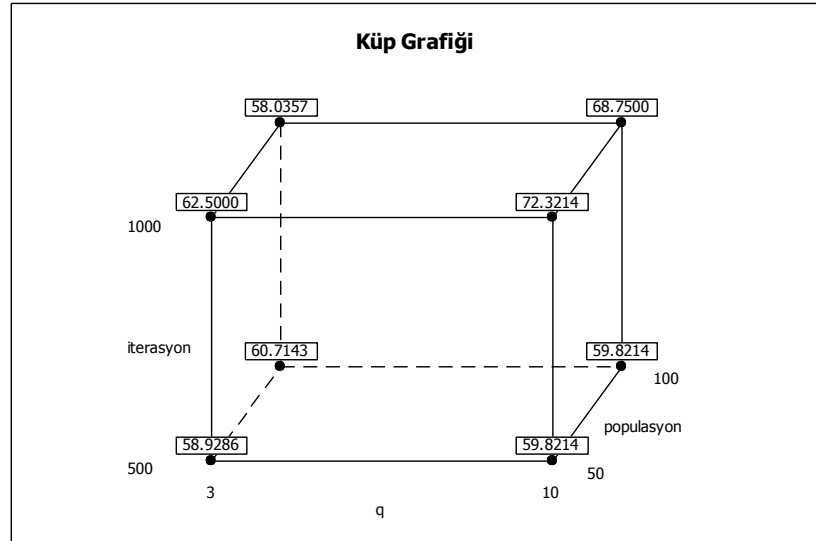
Ek 55 SFS algoritması Soybean veri kümesi pareto grafiği



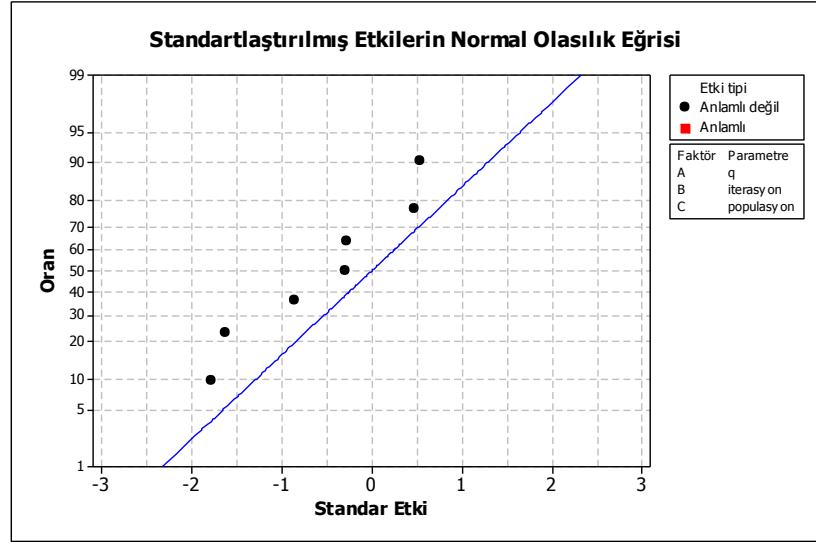
Ek 56 SFS algoritması Soybean veri kümesi tekli parametre etkileri



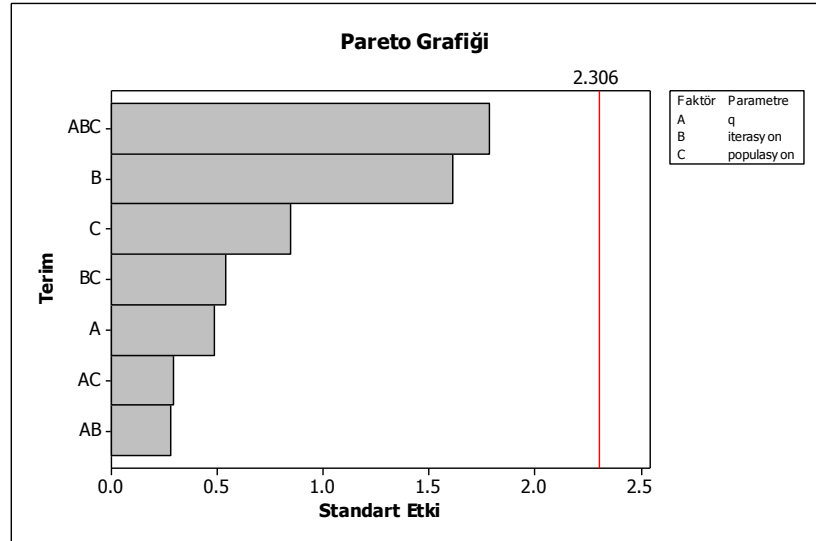
Ek 57 SFS algoritması Soybean veri kümesi ikili parametre etkileşimleri



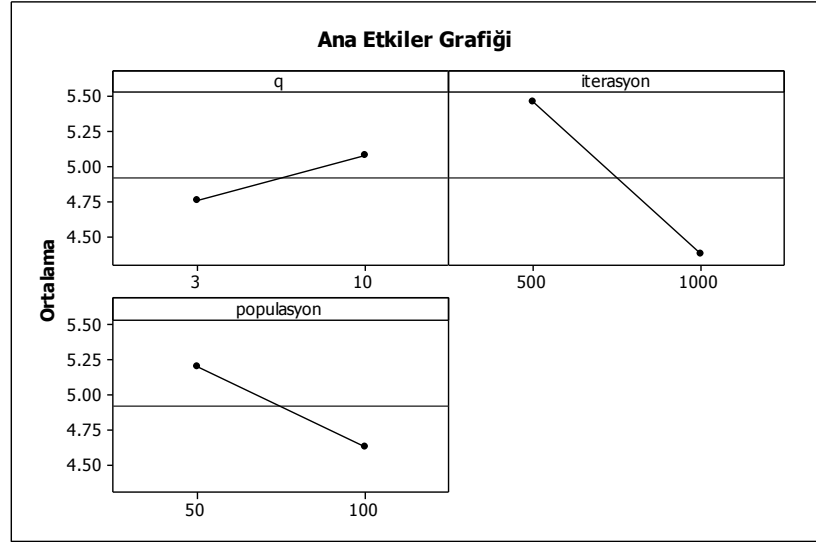
Ek 58 SFS algoritması Soybean veri kümesi küp grafiği



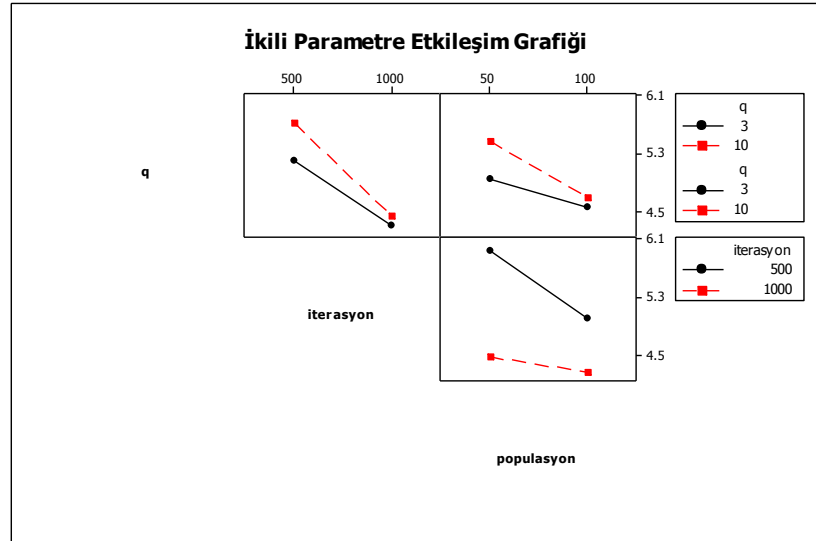
Ek 59 SFS algoritması Thyroid veri kümesi normal olasılık eğrisi



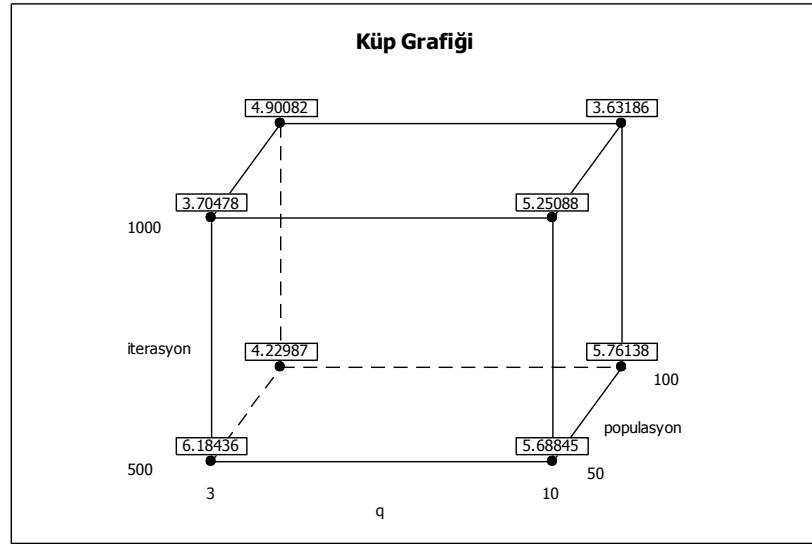
Ek 60 SFS algoritması Thyroid veri kümesi pareto grafiği



Ek 61 SFS algoritması Thyroid veri kümesi tekli parametre etkileri



Ek 62 SFS algoritması Thyroid veri kümesi ikili parametre etkileşimleri



Ek 63 SFS algoritması Thyroid veri kümesi küp grafiği

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı ve Soyadı : Burak Gülmez
 Uyuğu: Türkiye Cumhuriyeti
 Doğum Tarihi ve Yeri: 21.10.1991 / Sivas
 Medeni Durumu: Bekar
 Telefon Numarası: +90 505 829 50 95
 Email: burakgulmez@erciyes.edu.tr
 Adres: Erciyes Üniversitesi Mühendislik Fakültesi Endüstri
 Mühendisliği Melikgazi / Kayseri

EĞİTİM

Derece	Okul	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi	2016
Yüksek Lisans	Hacettepe Üniversitesi	2015 (Yatay Geçiş)
Lisans	Hacettepe Üniversitesi	2014

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2014 – Halen	Erciyes Üniversitesi	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

Gülmez B., Kulluk S., Özbakir L., "Karınca Koloni Algoritması Kullanılarak Su Dağıtım Sisteminin Optimizasyonu", Yöneylem Araştırması ve Endüstri Mühendisliği Kongresi, ANKARA, TÜRKİYE, 9-11 Eylül 2015, ss.169-169

Gülmez B., Kulluk S., "İyon Hareketi Algoritması ile Yapay Sinir Ağları Eğitimi", Yöneylem Araştırması ve Endüstri Mühendisliği Kongresi, İZMİR, TÜRKİYE, 13-15 Temmuz 2016, ss.120-121

Gülmez B., Kulluk S., "Training Artificial Neural Networks with Social Spider Algorithm", Üretim Araştırmaları Sempozyumu, İSTANBUL, TÜRKİYE, 12-14 Ekim 2016